



**KERNFORSCHUNGSANLAGE JÜLICH GmbH**

**Institut für Festkörperforschung**

**Zentrallabor für Elektronik**

**Rechnergesteuerte Anlage zur Synthese  
und Untersuchung von Metallwasser-  
stoff Systemen  
Automatische Be- und Entgasungs  
Anlage A B E A**

von

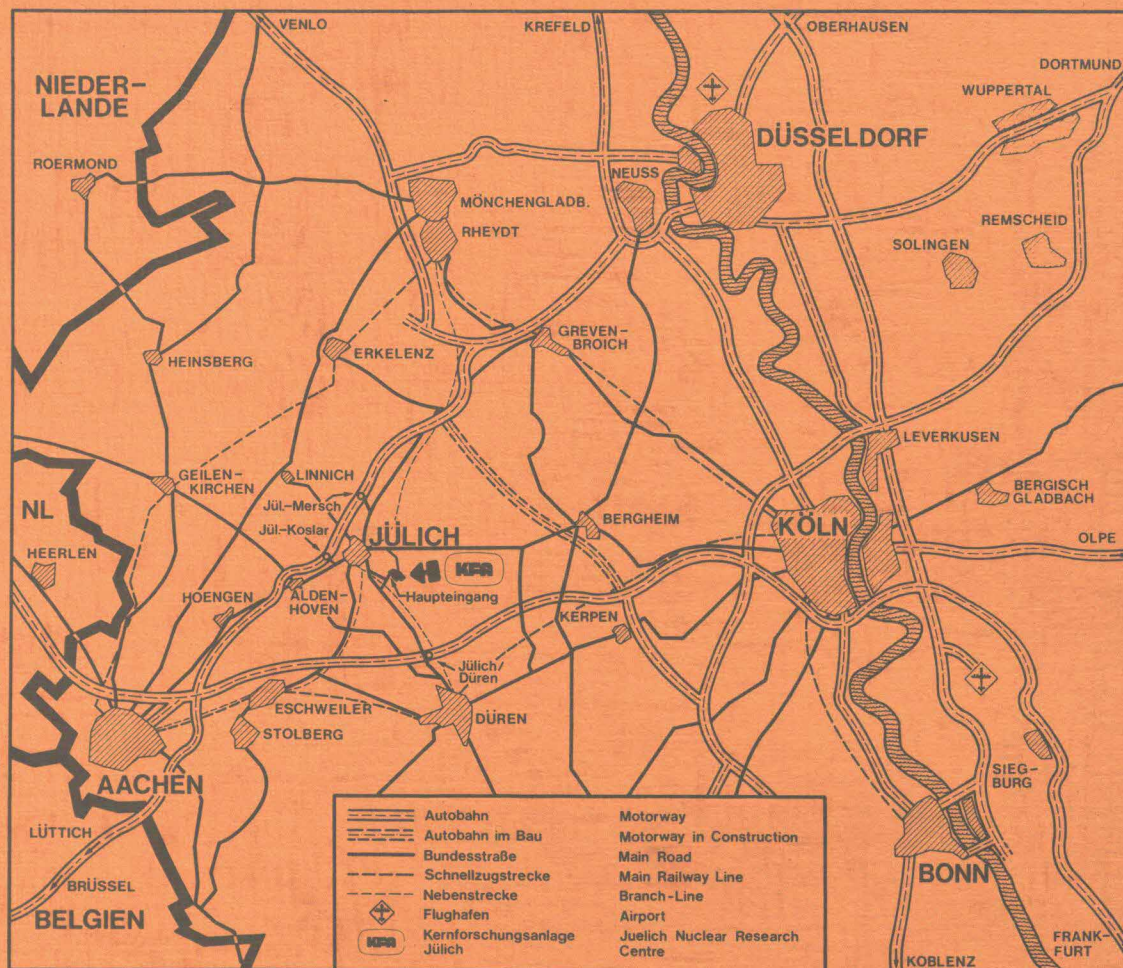
H. Heer, K. H. Klatt und P. Mecking

**Jül - Spez - 83**

**Juni 1980**

ISSN 0343-7639





Als Manuskript gedruckt

### Spezielle Berichte der Kernforschungsanlage Jülich - Nr. 83

Institut für Festkörperforschung Jül - Spez - 83  
Zentrallabor für Elektronik

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH  
Postfach 1913 · D-5170 Jülich (Bundesrepublik Deutschland)  
Telefon: 02461/611 · Telex: 833556 kfa d

**Rechnergesteuerte Anlage zur Synthese  
und Untersuchung von Metallwasser-  
stoff Systemen  
Automatische Be- und Entgasungs  
Anlage A B E A**

von

H. Heer \*, K. H. Klatt \*\* und P. Mecking \*\*

\*) Zentrallabor für Elektronik, KFA Jülich

\*\*) Institut für Festkörperforschung, KFA Jülich

INHALT  
\*\*\*\*\*

Seite

	Zielsetzung	1
1	Einleitung	2
2	Aufbau der Apparatur	4
2.1	Probenvolumen und Vorvolumina	4
2.2	Expansionsvolumen	4
2.3	Massenspektrometrie	5
2.4	Abpumpen der Anlage	5
2.5	Tritium - Anlage	6
3	Mess- und Steuergeraete	7
3.1	Ventile	7
3.2	Druckmesszellen	7
3.3	Temperaturmessung	7
3.4	Temperaturregelung	8
3.5	Digitalvoltmeter	8
3.6	Messstellenumschalter	8
3.7	Vordruckmessung	8
3.8	UHV-Druckmessung	8
4	Rechnerkonfiguration	9
5	Experiment - Rechner : C A M A C	10
6	Betriebssystem	12
6.1	Systemgenerierung	13
6.2	BOOT - Vorgang und STARTUP Aktivitaeten	15
7	SOFTWARE	17
7.1	C A M A C - Software	17
7.1.1	Erstellung der EXTERNAL PAGE	18
7.2	Beschreibung der CAMAC TASK	19
7.3	CAMAC Interrupt - Routinen	27
7.4	CAMAC Initialisierung	31
7.5	Modul Initialisierung	32
8	HP 2647/A Software	33
8.1	OFF LINE Software	33
8.2	PDP Software zur Vorbereitung des Terminals	34
8.3	PDP Software zur Terminal - Printer Funktion	39
8.4	Terminal Kassetten Software CIP	42
8.5	ON - OFF LINE - HP BASIC Programm (Graphik)	44
9	FORTTRAN C A M A C CALLS	46
10	Programmentwicklung	47
10.1	Ebene I	47
10.2	Ebene II	47
10.3	Beispiele	48

11	Datenverwaltung	50
11.1	COMMON Bereiche	50
11.1.1	Erstellung	50
11.1.2	Installierung	50
11.1.3	Initialisierung	50
11.1.4	Datenverwaltung im COMMON	51
11.1.5	Protokollierung der COMMON Daten	51
11.2	DIRECT ACCESS FILES	52
11.2.1	Zweck	52
11.2.2	Erstellung	52
11.2.3	Initialisierung	53
11.2.4	Datenverwaltung in den FILES	53
12	Software - Synchronisierung	54
13	Abbild des Experiments im COMMON III	55
14	Beispiel einer vorbereitenden Messung	57
15	Beispiel eines Mess- und Steuerprogramms	59
16	Test - Software	69
16.1	ADC Test	69
16.2	DAC Test	69
16.3	Parallel Output Register Test (Input - Reg.)	70
16.4	Messstellenumschaltung	70
16.5	Ventilsteuerung	70
16.6	Experimentsimulation	71
16.7	Test : Flag - Verwaltungs	71
16.8	Test : HP Graphic - BASIC	71
17	Unterprogrammbibliothek ABEA.OLB (alphabetische Reihenfolge)	72
18	Ausblick	79
19	Fotografien	80



## Zielsetzung

Um Untersuchungen an Metall-Wasserstoff Systemen durchfuehren zu koennen, benoetigt man eine Apparatur, in der man die zu untersuchenden Proben (meist Folien oder kleine feste Stuecke) aufheizen und mit definierten Druucken von Wasserstoff beaufschlagen kann. Die durch die Gasaufnahme bedingten Druckabfaelle muessen ebenso wie die Temperatur gemessen werden.

Aussrdem muss die Apparatur zu Beginn der Messungen in einen Zustand hoechster Reinheit versetzt werden koennen d.h. mit Hilfe von Pumpen muss ein Vakuum von mindestens  $10^{-7}$  mbar erreicht werden, um Verunreinigungseffekte auszuschliessen.

Bei der hier verwendeten Apparatur handelt es sich also um eine Ultrahochvakuumapparatur, die gleichzeitig fuer Gasdruecke bis 70 bar ausgelegt ist.

Die im IFF bisher gebauten Apparaturen dieser Art wurden manuell bedient und gesteuert. Um nun Untersuchungen an einer Vielzahl von Proben vornehmen zu koennen, wurde der Bau einer Anlage in Angriff genommen, bei der, angeschlossen an einen Prozessrechner, Mess- Steuerungs- und Auswertungsvorgaenge vollautomatisch ablaufen sollten.

Dies beinhaltet im wesentlichen drei Vorgaenge:

- a) Regelung und Messung der Temperatur der Probe
- b) Regelung und Messung des Gasdrucks
- c) Steuerung (d.h. Oeffnen und Schliessen) der Ventile
- d) Auswertung (zum Teil graphische) der Messergebnisse

## 1 Einleitung

Schon seit mehreren Jahren liegt ein Forschungsschwerpunkt des Instituts fuer Materialentwicklung des IFF in der Beschaeftigung mit Metall-Wasserstoff-Systemen.

Dies sind Metalle bzw. Legierungen, die den Wasserstoff auf Zwischengitterplaetzen, d.h. auf Plaetzen zwischen den regulaeren Gitteratomen des Festkoerpers einlagern koennen. Da die Speicherung des Wasserstoffs teilweise in recht grossen Mengen und unter thermodynamisch guenstigen Bedingungen erfolgen kann, bietet sich die Verwendung dieser Materialien als technisch interessante Energiespeicher an.

Im Hinblick auf die technische Nutzbarkeit dieser Metall-Wasserstoffsysteme muessen nun zahlreiche Stoffe auf ihre Beladbarkeit und ihre thermodynamischen Eigenschaften bezueglich der Wasserstoffeinlagerung untersucht werden.

Dabei muessen im wesentlichen drei Eigenschaften geprueft werden:

- a) Die Speicherdichte des Wasserstoffs.  
(In einigen Metallen kann die Dichte des gespeicherten Wasserstoffs die der fluessigen Phase uebersteigen.)
- b) Die Randbedingungen, unter denen Wasserstoff in den Speicher eingelagert werden kann.  
Dies sind im wesentlichen Druck und Temperatur.  
Man will also wissen, welche  $H_2$ -Konzentration man bei bestimmten Druecken und bei bestimmten Temperaturen erreicht.  
Eine typische Messung ist hier die Bestimmung der sogenannten Druck-Konzentrations Isothermen, also Kurven gleicher Temperatur, bei denen die aufgenommene Wasserstoffkonzentration in Abhaengigkeit vom herrschenden aeusseren Gasdruck aufgetragen ist. Ein typischer Verlauf ist hier fuer die technisch interessante Legierung FeTi angegeben. (Abb. 1)

Die typische Druck-Konzentrations Isotherme, wie sie hier gezeigt ist, zeichnet sich durch folgende Eigenschaften aus: Der erste ansteigende Ast der Kurve bei kleinen Konzentrationen kennzeichnet eine Phase, in der Wasserstoff ungeordnet und statistisch auf den vorhandenen Zwischengitterplaetzen verteilt ist. Hier gilt oft das "Sievert-gesetz", das besagt, dass die Wasserstoffkonzentration proportional zur Wurzel des aeusseren Drucks ansteigt.

Das zweite Teilstueck, das sogenannte "Plateau" zeigt, dass hier ein Phasengleichgewicht zwischen zwei Phasen besteht: Der ersten ungeordneten Phase und einer zweiten Phase, in der die Wasserstoffatome untereinander durch ihre Wechselwirkung eine Ordnung

ausbilden. Diese beiden Phasen stehen hier im Gleichgewicht, was zur Folge hat, dass sich bei steigender Gaskonzentration der Gleichgewichtsdruck nicht ändert. Mit zunehmender Konzentration nimmt der Anteil der geordneten Phase zu, bis nur noch diese Phase vorhanden ist. An diesem Punkt endet das "Plateau" und die Kurve steigt wieder an. Bei weiter zunehmender Konzentration koennen sich noch weitere Plateaus ausbilden, die wiederum durch das Gleichgewicht zweier Phasen bestimmt sind.

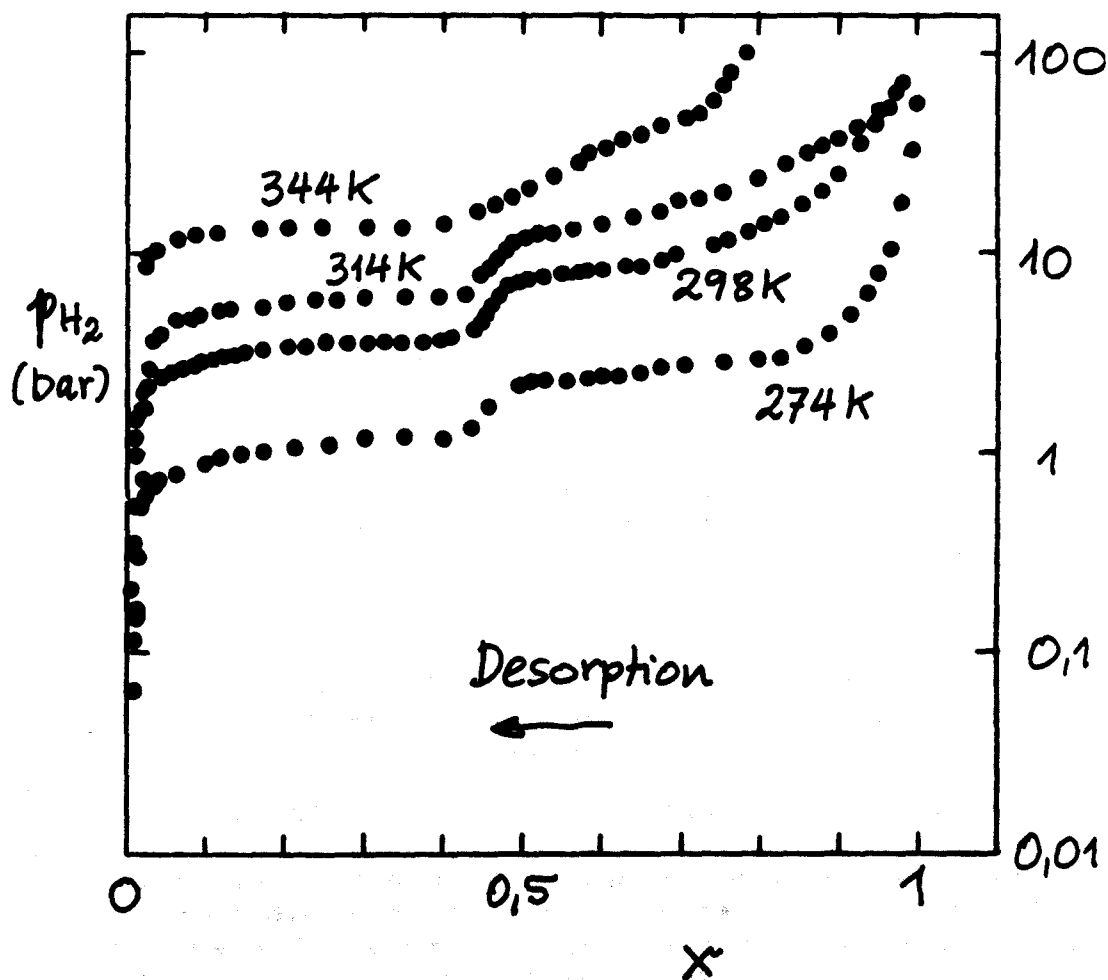


Abb. 1: Isothermen verschiedener Temperaturen von FeTi

- c) Einige Wasserstoffspeicher zeichnen sich dadurch aus, dass sie die Isotope des Wasserstoffs (Protium, Deuterium und Tritium) unterschiedlich gut speichern, d.h. ihr Druck-Konzentrations-Temperatur Verhalten fuer die drei Isotopen unterschiedlich ist. Dies bietet eine Moeglichkeit zur Trennung der Isotopen voneinander, was fuer verschiedene technische Anwendungen, so z.B. fuer die Fusionsreaktortechnologie wichtig ist.



## 2 Aufbau der Apparatur

### 2.1 Probenvolumen und Vorvolumina

In Abb. 2 ist eine Schemazeichnung der aufgebauten Anlage dargestellt.

Zentralstueck des Aufbaus ist die Beladungszelle (1). Es handelt sich um ein mit einem CF 35 Doppelschneidenflansch ultrahochvakuumdicht abgeschlossenes Volumen, bestehend aus einem V2A-Rohr, das doppelwandig ausgefuehrt ist.

Hierin befindet sich, am Flansch befestigt, ein Quarzroehrchen, das als Probenbehaelter dient. (Abb. 3)

Zur Temperaturmessung und -regelung sind zum Aussenraum zwei NiCr-Ni Thermoelemente durchgefuehrt. Die Heizung des Probenraums erfolgt ueber einen Molybdaendraht, der um das Quarzrohr gewickelt ist.

Die doppelwandige Ausfuehrung der Beladungszelle dient dazu, diese durch Wasserdurchfluss zu kuehlen, und so zu verhindern, dass sich die restliche Apparatur langsam durch Waermeleitung aufheizt.

Um den Gasdruck in der Beladungszelle zu messen, ist ein kapazitiver Absolutdruckaufnehmer der Fa. SETRA, Serie 204 am Beladungsvolumen angebracht.

Das Probenvolumen und die in der Abbildung 2 stark durchgezogen dargestellten Volumina sind bis zu einem Maximaldruck von 70 bar ausgelegt. Die Absolutdruckaufnehmer sind ebenfalls fuer Druckmessungen bis 70 bar geeignet.

Das Probenvolumen ist ueber einen CF 16 Doppelschneidenflansch an die Restapparatur angeschlossen. Das Ventil V7 dient zum Abschluss der Beladungszelle von der Restapparatur.

Um nun Wasserstoff (H<sub>2</sub>) oder Deuterium (D<sub>2</sub>) in die Beladungszelle einlassen zu koennen, muss in die Vorvolumina die sich zwischen den Ventilen V4 und V5 bzw. V1 und V2 befinden, ein bestimmter Anfangsdruck eingelassen werden. Die Vorvolumina bestehen aus 12x1 mm V2A-Rohren, und haben jeweils ein Volumen von etwa 33 cm<sup>3</sup>. Hieran angebracht sind wiederum jeweils ein kapazitiver Absolutdruckaufnehmer (SETRA Serie 204).

Der in den Vorvolumina aufgebaute Druck kann ueber die Ventile V2 bzw. V5 und V6 und V7 in das Probenvolumen eingelassen werden. Die Ventile sind durch 12x1 mm Rohre aus V2A-Stahl miteinander verbunden.

### 2.2 Expansionsvolumen

Um andererseits schon mit H<sub>2</sub> oder D<sub>2</sub> beladene Proben durch Aufheizung wieder zu entgasen, und den dabei entstehenden Druckanstieg zu messen, ist ein Expansionsvolumen von ca. 1l Groesse angebracht worden. Dies ist deshalb so

gross gewaehlt worden, um bei einer Entgasung der Probe den Druckanstieg moeglichst klein zu halten, sodass die Probe fast ihre gesamte Wasserstoffmenge abgeben kann, und erst bei sehr kleiner Konzentration ein Gleichgewicht mit dem umgebenden Gas entsteht. So ist es moeglich, durch ein- oder zweimaliges Ausgasen der Probe in das evakuierte Expansionsvolumen den gesamten Wasserstoffinhalt der Probe auszutreiben. Der Druckanstieg im Expansionsvolumen wird mit Hilfe eines hochgenauen Kapazitaetsmanometers (Messbereich 0 - 100 Torr) bestimmt. Es handelt sich um eine Absolutdruckmesszelle der Fa. Datametrics Typ Barocel 572 A. Das Expansionsvolumen kann ueber das Ventil V12 mit der Beladungszelle verbunden werden.

### 2.3 Massenspektrometrie

In der zweiten Ausbaustufe ist vorgesehen, an die Beladungs- und Analysenapparatur ein Massenspektrometer anzuschliessen. Hiermit soll bei Beladungszyklen mit Gasgemischen die quantitative Gaszusammensetzung des Restgases ueber der Probe ermittelt werden.

Zu diesem Zweck ist es moeglich, ueber das Ventil V13 aus dem Probenvolumen eine kleine Menge an Gas zu entnehmen. Das Volumen zwischen V13 und V14 ist relativ klein ( $7 \text{ cm}^3$ ) sodass der Druckabfall im Probenvolumen nicht zu gross ist. Diese Gasprobe, die maximal einen Druck von 70 bar aufweisen kann, muss um in das Massenspektrometer eingelassen werden zu koennen, auf einen Druck zwischen 0 und 10 mbar gebracht werden.

Hierzu kann die Gasprobe bedarfsweise ueber die Ventile V14 V15 und V16 in ein oder zwei grosse Expansionsvolumina gelassen werden, wodurch der Druck soweit abgesenkt werden kann, dass ein Einlass in das Massenspektrometer moeglich wird.

### 2.4 Abpumpen der Anlage

Um die gesamte Anlage oder Teile derselben in einen Ultrahochvakuumzustand versetzen zu koennen ( $p < 10^{-8} \text{ mbar}$ ) ist ein Turbomolekularpumpstand der Fa. Pfeiffer (Saugvermoegen 110 l/sec) ueber das Ventil V8 bzw. V9 mit der Anlage verbunden. Die Pumpe ist ueber das Ventil V9 direkt und ueber das Ventil V8 ueber eine Blende von 1mm Durchmesser mit an die Anlage angeschlossen. Die Blende dient dazu, bei hohen Druecken ( $p > 5 \text{ bar}$ ), die Durchflussrate zu verringern, und so die Turbopumpe zu schonen.

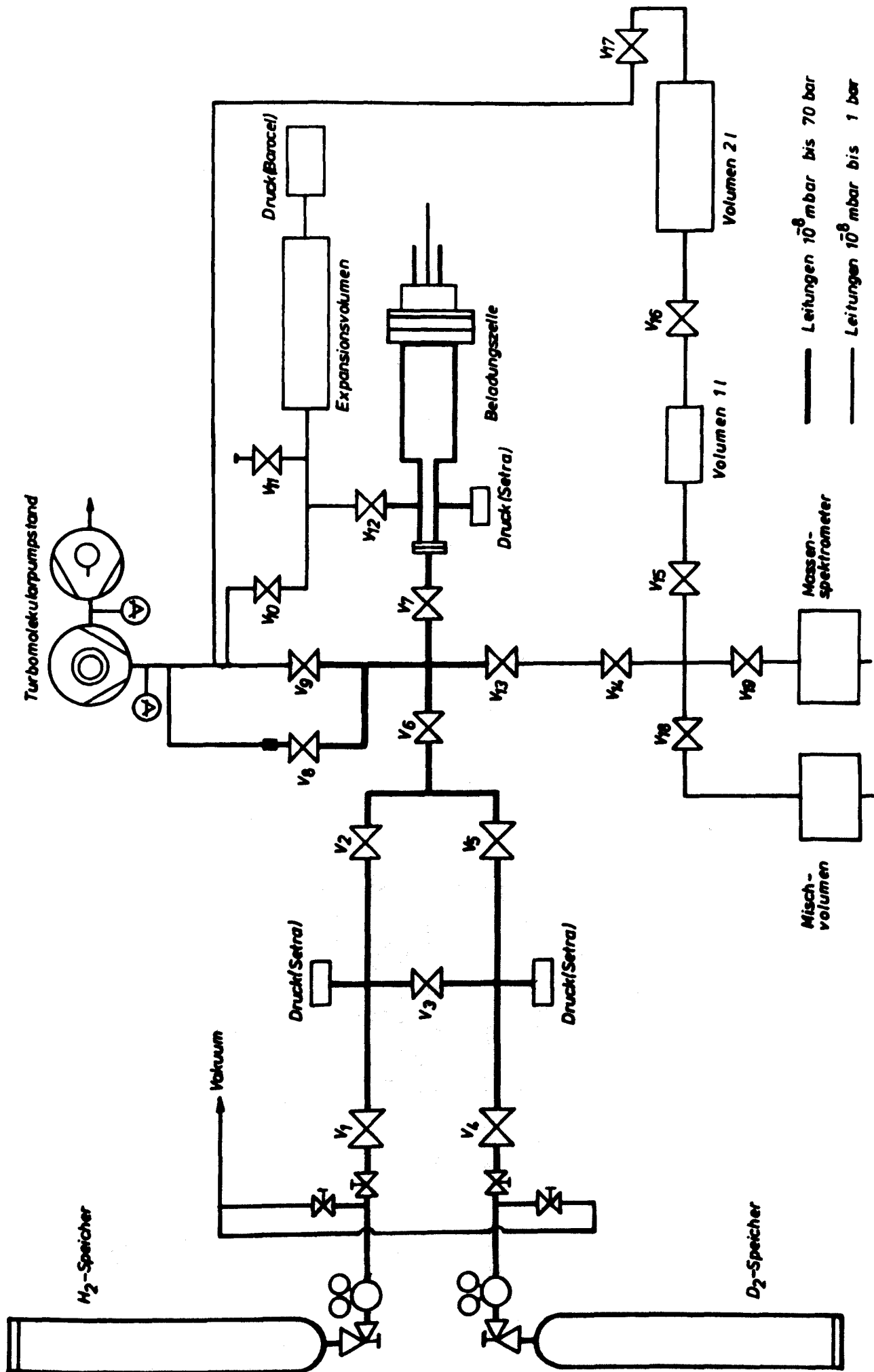


Abb. 2: Schemazeichnung der ABEA - Apparatur

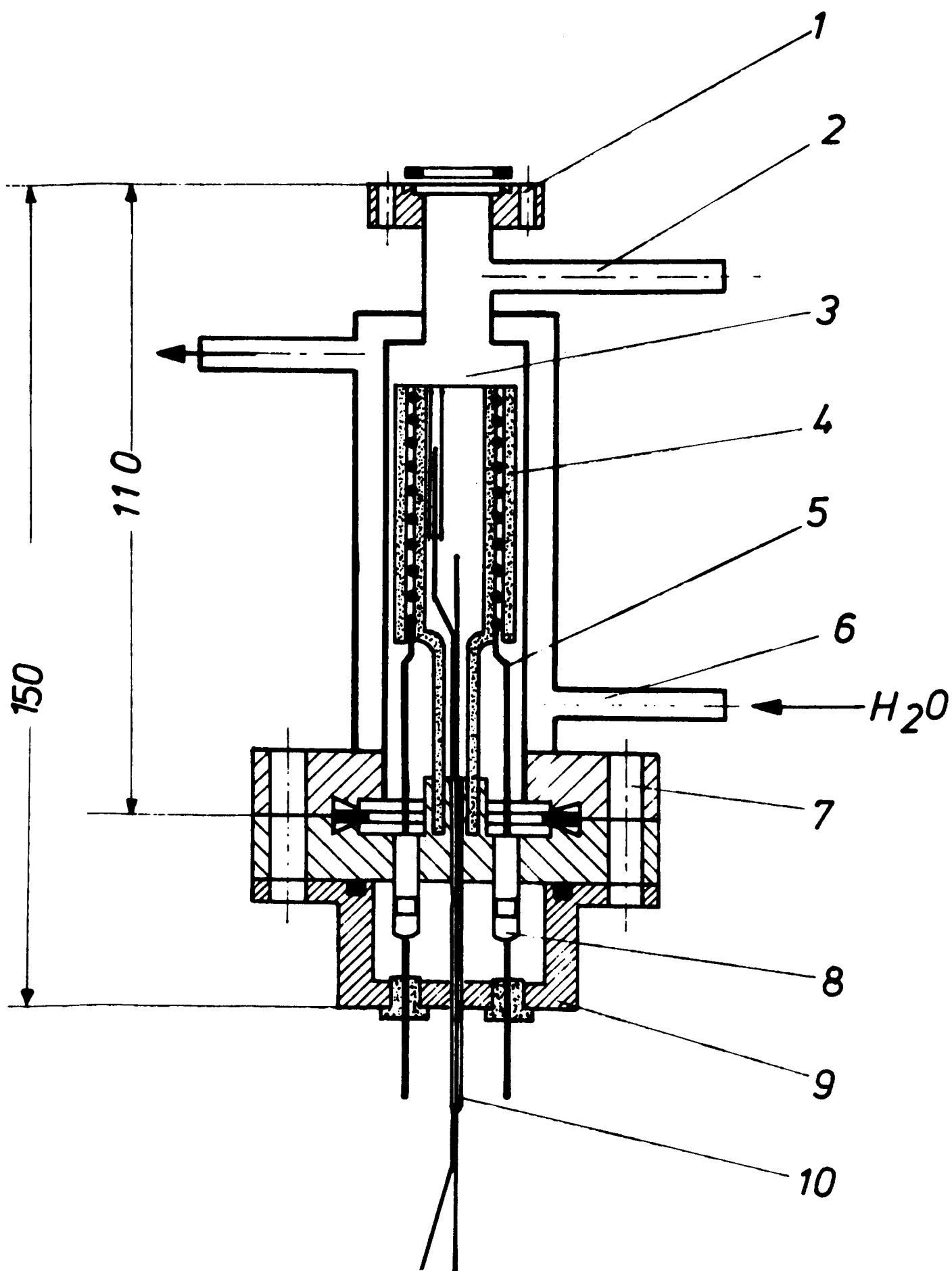


Abb. 3: Beladungszelle



## 2.5 Tritium-Apparatur

Eine zweite UHV-Apparatur, in der mit Tritium experimentiert wird, ist ebenfalls an den Rechner angeschlossen. Da diese Anlage in einem weit entfernten Labor im Kontrollbereich aufgebaut ist, musste der Anschluss ueber eine 150 m lange Kabelstrecke durchgefuehrt werden, was jedoch problemlos gelang.

Bei dieser Apparatur handelt es sich um einen der oben beschriebenen Anlage sehr aehnlichen Aufbau, wobei jedoch als einziger Unterschied die automatische Betaetigung der Ventile entfaellt. Die Steuerung durch den Rechner beschraenkt sich also auf Aenderungen der Temperatur und Messung von Druck und Temperatur. Auch hierdurch wurden jedoch schon grosse Zeitgewinne verzeichnet.

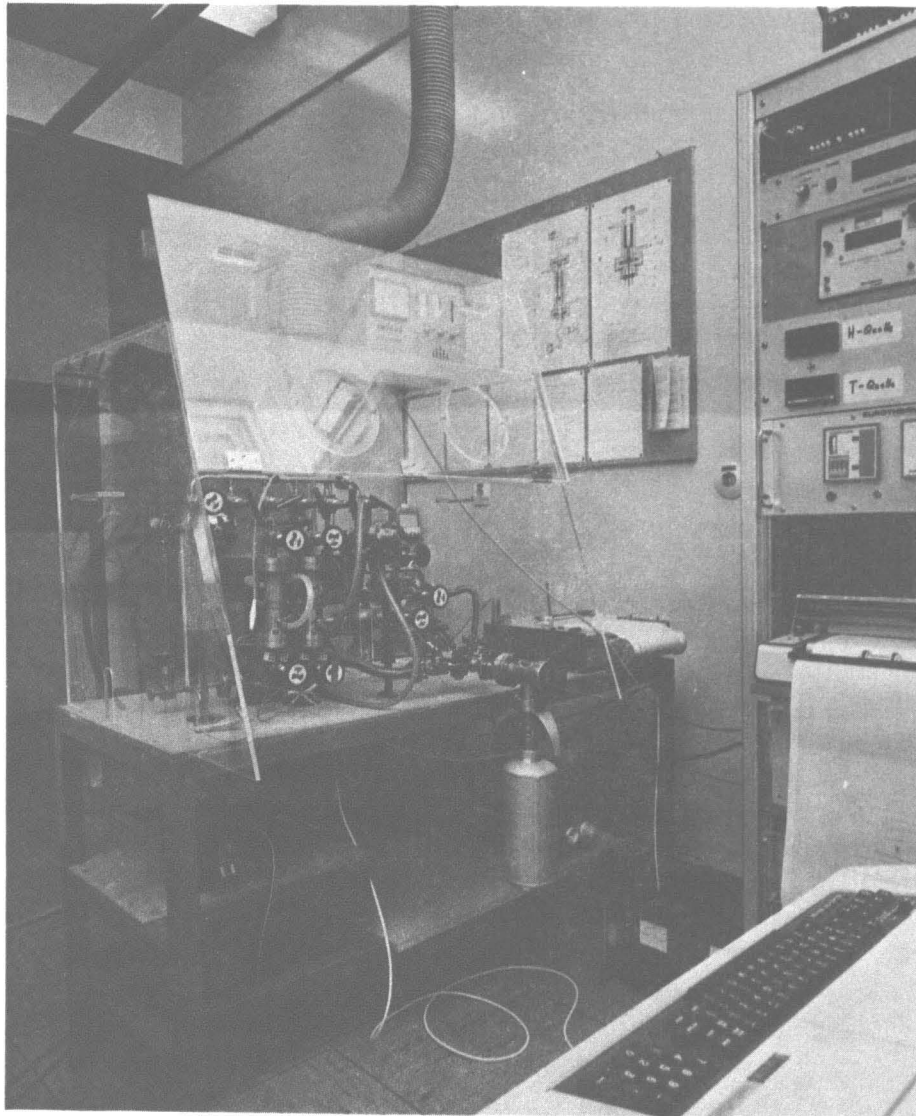


Abb. 4: Fotografie der Tritium-Apparatur

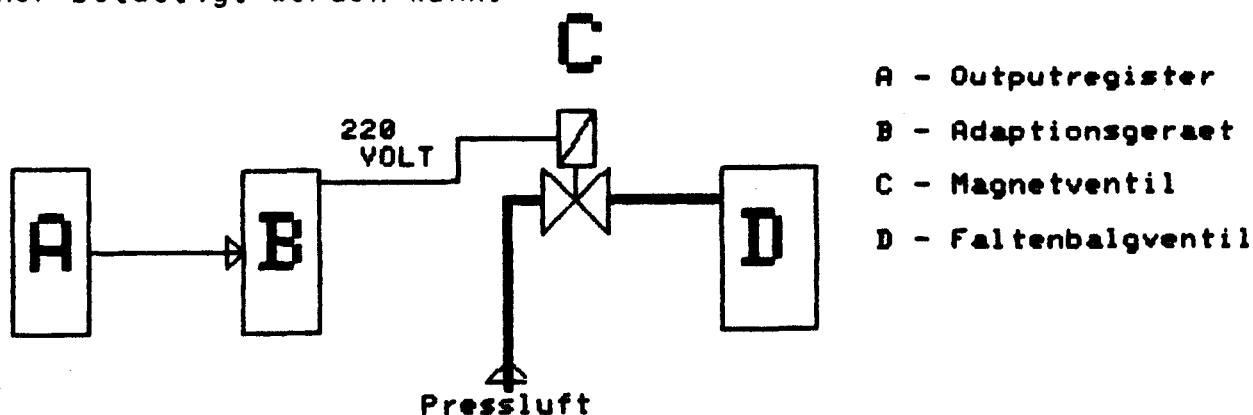
### 3 Mess- und Steuergeraete

#### 3.1 Ventile

Die verwendeten Ventile der Fa. NUPRO sind pneumatisch betätigte UHV-dichte Auf-Zu Faltenbalgventile, die fuer einen Maximaldruck von 70 bar ausgelegt sind.

Das Schliessen dieser Ventile geschieht durch Federkraft, geöffnet werden sie durch Druckbeaufschlagung mit 6 bar Pressluft. Die notwendige Pressluft zum Öffnen eines Ventils wird aus dem Hausnetz entnommen und ueber ein Magnetventil der Fa. Asco Typ K8 320 A1 dem entsprechenden Faltenbalgventil zugeführt.

Die Magnetventile werden mit Hilfe eines im Zentrallabor fuer Elektronik gebauten Adaptionsgeraetes mit der notwendigen Schaltspannung von 220 V versorgt. Das Adaptionsgeraet ist an ein TELEFUNKEN Output Register Typ MS P0 2 1230/1 angeschlossen, sodass ueber CAMAC jedes einzelne Ventil vom Rechner betätigt werden kann.



#### 3.2 Druckmesszellen

An den Vorvolumina sowie an der Beladungszelle sind kapazitive Absolutdruckmesszellen der Fa. SETRA Serie 204 angebracht. (Druckbereich 0-1000 PSI  $\Leftrightarrow$  0-69 bar)

Diese Druckmesszellen liefern proportional zum gemessenen Druck eine Gleichspannung von 0-10 Volt mit einer Genauigkeit von 0,01 PSI = 0,7 mbar.

Der Druck im Expansionsvolumen wird durch ein hochgenaues Kapazitaetsmanometer der Fa. Datametrics Typ BAROCEL 572 A (Messbereich 0-100 Torr) gemessen. Das Geraet liefert proportional zum anliegenden Druck eine Gleichspannung von 0-10 Volt mit einer Genauigkeit von 0,01 Torr.

#### 3.3 Temperaturmessung

Die Temperatur der Probe wird durch ein NiCr-Ni Thermoelement gemessen und mittels eines Temperaturmess- und Anzeigegeraetes der Fa. MAWITHERM Typ 3751 angezeigt. Dieses Geraet liefert proportional zur Temperatur eine Analogspannung von 0-10 Volt. (1 mV/ 0,1 deg C) mit einer Genauigkeit von etwa 1%.

### 3.4 Temperaturregelung

Das Heizen des Probenvolumens erfolgt, wie in Kap. 2.1 beschrieben, durch einen auf ein Quarzglasroehrchen gewickelten Molybdaendraht. Die Heizleistung wird mit Hilfe eines PID-Reglers der Fa. EUROTHERM Typ 093 geregelt. Dieser Regler erhaelt als Regelinformation die Spannung eines NiCr-Ni - Thermoelements, das in der Naehе der Heizwicklungen befestigt ist (Siehe Abb. 3). Der Regler kann extern durch die Vorgabe einer Gleichspannung im Bereich 0-10 Volt programmiert werden. (0-10 Volt  $\Leftrightarrow$  0-1000 deg C)

### 3.5 Digitalvoltmeter

Um die Gleichspannung der vier verwendeten Kapazitaetsdruckmesszellen genau zu messen, ist ein 5 1/2 stelliges Digitalvoltmeter der Fa. RACAL DANA Typ 5100 installiert. Das Geraet besitzt einen BCD-Datenausgang.

### 3.6 Messstellenumschalter

Die vier zu messenden Gleichspannungen der Druckmesszellen sind ueber einen Messstellenumschalter der Fa. BURSTER Typ 1634 an das DANA-Digitalvoltmeter angeschlossen. Der Messstellenumschalter hat 24 Schaltmoeglichkeiten und ist "modulo 4" verdrahtet, d.h. bei fortlaufender Umschaltung ist die Messreihenfolge auf dem Digitalvoltmeter: 1,2,3,4,1,2,3,4,...1,2,3,4. Der Messstellenumschalter laesst sich durch kurze Gleichspannungspulse (20 msec, 5V) extern schrittweise umschalten.

### 3.7 Vordruckmessung

Die Messung des Vordrucks an der Vorpumpe erfolgt mit Hilfe einer Pirani-Messroehre der Fa. BALZERS Typ 010 mit zugehoerigem elektronischen Konvertierungsgeraet. Das Prinzip der Pirani-Messroehre beruht auf der Druckabhaengigkeit der Waermeleitfaehigkeit von Gasen. Das Anzeigegeraet liefert fuer den Anzeigebereich von  $10^{*-4}$  -  $10^0$  mbar eine Analogspannung von 0-10 Volt.

### 3.8 UHV-Druck Messung

Druecke, die kleiner als  $10^{*-4}$  mbar sind, werden ueber der Turbomolekularpumpe mit Hilfe einer Bayard-Alpert Roehre und zugehoerigem elektronischen Anzeige- und Auswertegeraet der Fa. BALZERS Typ IMG 070 gemessen. Das Geraet liefert proportional zum Druck von  $10^{*-10}$  bis  $10^{*-4}$  mbar eine Analogspannung von 0-10 Volt.

Das Prinzip der Bayard-Alpert Roehre beruht auf folgendem Effekt:

Durch Stoss mit beschleunigten Elektronen, die durch Glueh-emission erzeugt worden sind, werden Gasatome ionisiert. Die Ionen werden durch eine Saugspannung auf einen Ionen-faenger gezogen, wodurch der Ionenstrom gemessen werden kann. Dieser ist unter bestimmten Vorraussetzungen proportional dem Gasdruck an der Roehre.



4 Rechner - Konfiguration

PDP 11/34

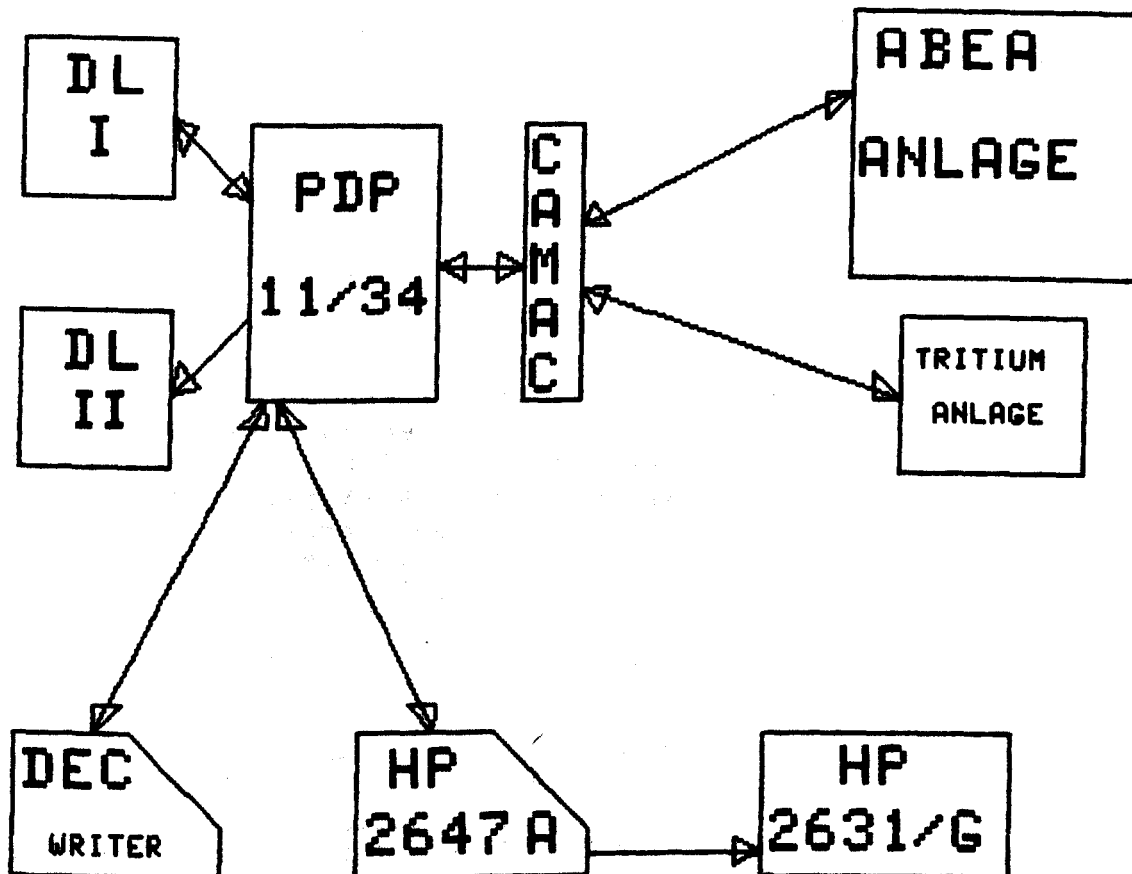
128 K Memory

2 DL01 Plattenlaufwerke

1 DEC - WRITER

1 Graphisches Display HP 2647/A (Hewlett\*Packard)

Diagramm der Rechnerkonfiguration  
\*\*\*\*\*

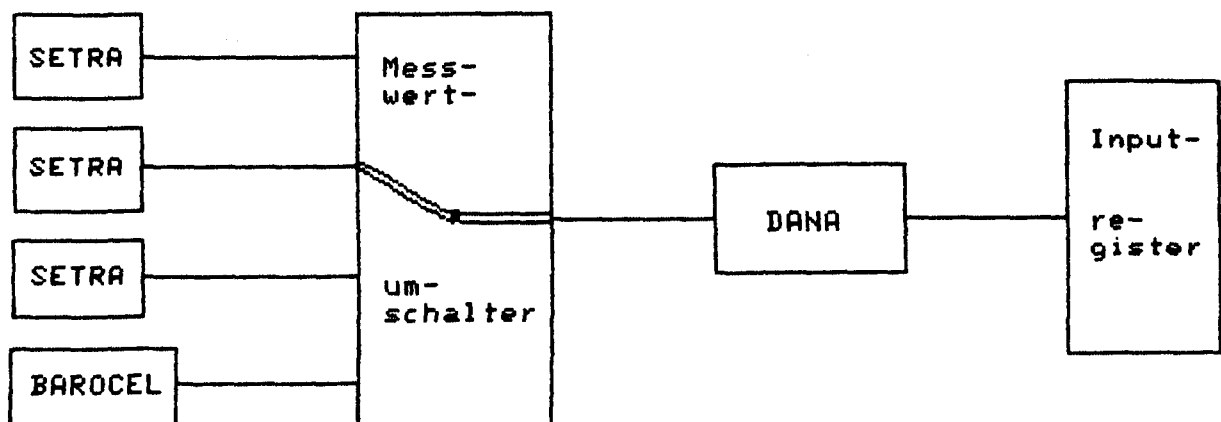


## 5 Experiment - Rechner - CAMAC

### 5.1 Druckmesszellen

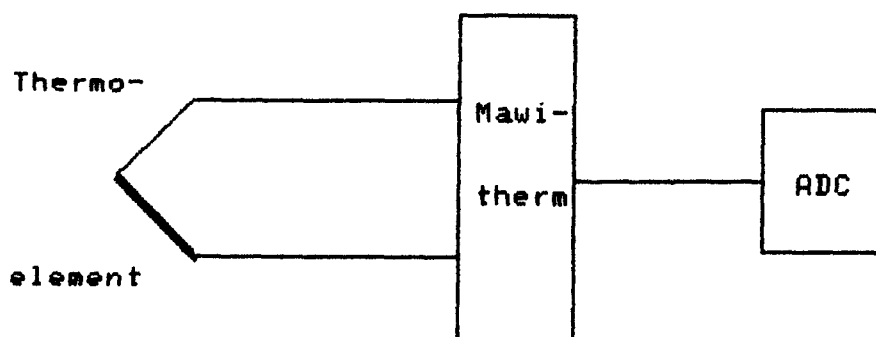
Die Ausgangsspannung der vier Kapazitätsmanometer (0-10 Volt) wird ueber den Messstellenumschalter auf das Digitalvoltmeter geleitet. Die vier Zellen koennen nacheinander durch Umschalten des Messstellenumschalters abgefragt werden.

Das Digitalvoltmeter besitzt einen BCD-Datenausgang, der mit dem TELEFUNKEN Parallel Input Register Typ MS PI 1 1230/1 verbunden ist. Die Daten werden im Abstand von ca. 300 msec uebernommen und abgelegt.



### 5.2 Temperaturmessung

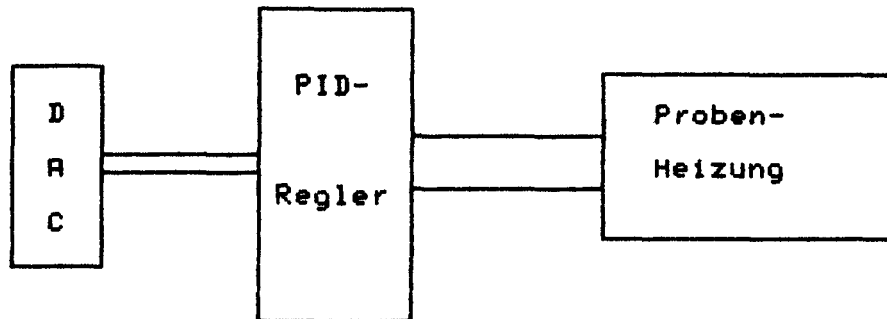
Das Messthermoelement (NiCr-Ni), das die Probertemperatur misst, ist an ein Temperaturmessgeraet der Fa. MAWITHERM Typ 3751 angeschlossen. Dieses Geraet liefert fuer einen Temperaturbereich von 0-1000 deg C eine Analogspannung von 0-10 Volt (1 mV/0,1 deg C). Diese Analogspannung ist direkt mit einem der Eingaenge des JOERGER ADC Typ AMI verbunden.



### 5.3\* Temperaturregelung

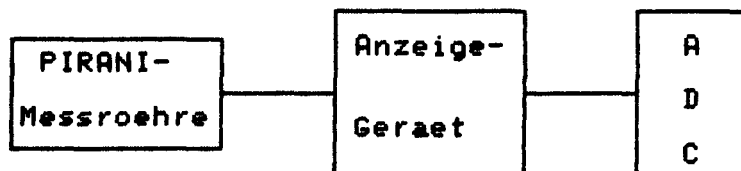
Die Heizung des Probenvolumens erfolgt ueber einen PID-Regler der Fa. EUROTHERM Typ 093.

Der Regler erhaelt vom KINETICS DAC Typ 3112 eine Vorgabespannung im Bereich von 0-10 Volt und regelt proportional zu dieser Spannung eine Temperatur von 0-1000 deg C ein.



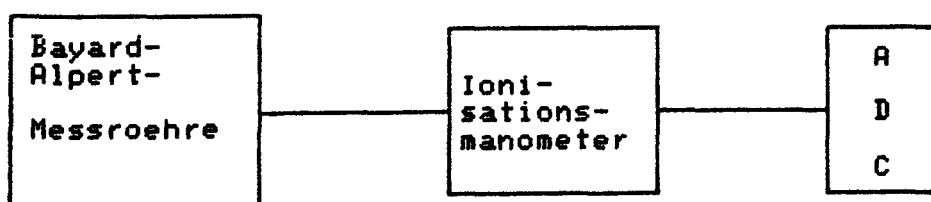
### 5.4 Vordruckmessung

Der von der Vorpumpe erzeugte Druck wird mit Hilfe einer Pirani Messroehre gemessen. Das Anzeigegeraet liefert eine Analogspannung von 0-10 Volt fuer den Druckbereich von  $10^{-4}$  bis 1000 mbar. Diese Spannung ist wiederum direkt am JOERGER ADC angeschlossen, und kann so ueber CAMAC ausgelesen werden.



### 5.5 UHV-Druck Messung

Der UHV-Druck wird ueber der Turbopumpe mittels einer Bayard-Alpert Roehre gemessen, die an ein Ionisationsmanometer der Fa. BALZERS Typ IMG 070 angeschlossen ist. Die von diesem Geraet gelieferte Analogspannung von 0-10 Volt fuer den Druckbereich von  $10^{-11}$  bis  $10^{-4}$  mbar ist direkt an den JOERGER ADC angeschlossen.



## 6 Das Betriebssystem RSX 11M V3.1

RSX 11/M V3.1 ist ein Real-Time Multiprogramming System fuer die PDP-11 Serie der DIGITAL Rechner. Es ist fuer schnelle Abarbeitung von Echtzeitanforderungen ausgelegt.

Auf der einen Seite kann RSX 11M als stand-alone Prozesssteuerungssystem ausgelegt werden, auf der anderen Seite als ein Mehrbenutzer System fuer Anwendungssoftwareentwicklung dienen.

Beim vorliegenden Experiment soll ein Mittelweg zwischen beiden Aufgabenbereichen gegangen werden. Programmentwicklung und Modifizierung bei gleichzeitiger Steuerung mehrerer Experimente. Zur Zeit sind zwei Experimente angeschlossen, es ist aber denkbar, weitere aehnlich strukturierte Experimente in Zukunft an den Rechner anzuschliessen.

Die Hauptgebiete einer Real-Time Anwendung sind die Daten-Aquisition, Prozesskontrolle und Datenanalyse.

Die Datenaquisition ist das Sammeln von Physikalischen Daten fuer spaetere Auswertung und Kontrollfunktionen. RSX 11M sieht eine direkte Abnahme der Daten durch aeusseren Anstoss vor weil das System bzw. die geschriebenen Programme fuer die Kontrolle der Externgeraete direkt an die PDP-11 - Hardware - Interruptvektoren gebunden sind.

Die Prozesskontrolle beinhaltet die Datenaquisition, deren Analyse und ein Feed- Back Kreis zum Experiment. Benutzer-Programme generieren Signale, die auf den Prozess Einfluss nehmen. RSX 11M kann Prozess- Inputs und Prozess- Outputs in einem gewaehrleisteten Echtzeitverhalten bearbeiten.

Die vom System angebotenen Programmiersprachen FORTRAN IV, BASIC und der MACRO11 Assembler sind fuer die gesamte Programmentwicklung fundamentale Hilfsmittel.

FORTAN IV : Programmsystem zur Datenerfassung und Steuerung

ASSEMBLER : Driver, Handler, Konvertierungsprogramme

BASIC : Testprogramme

Ein sehr wichtiger Faktor ist bei einem Multi-User System die Synchronisierung aller Aktivitaeten im Rechner. Gleichzeitig laufende Programme muessen zusaetzlich zu jedem Zeitpunkt eine Bearbeitung externer Interrupts ohne messbare Zeitverzoegerung zulassen.

RSX 11M bitet hierfuer das Synchronisationskonzept der EVENT-FLAGS und die Systemaufrufe wie SEND, RECEIVE, SET FLAG, DECL FLAG und WAIT FLAG an. Mit diesen Hilfsmitteln liess sich ein System aufbauen, bei dem jeder Benutzer, jedes Experiment den Eindruck gewinnen muss, allein vom Rechner jederzeit bedient zu werden. Bei der Programmentwicklung haben die RSX 11M UTILITIES erhebliche Arbeitserleichterung angeboten.

PIP>	EDI>	LBR>
MAC>	FOR>	BAS>
DMP>	ODT>	



## 6.1 Systemgenerierung

Bei der Systemgenerierung RSX 11M V3.1 wurde eine optimale Anpassung der Hardware- Konfiguration und der zu erwartenden Software- Belastung des Rechners angestrebt. Das Quell- Betriebssystem wurde auf drei Plattenspeichern (RL01) geliefert.

- a. Mapped System System Object Disc (MAPSYS)
- b. Utility Object Disc (RLUTIL)
- c. FORTRAN

Vor der Systemgenerierung wurden die Quellplatten mit folgenden Kommandos kopiert :

```
MCR>INS [1,54]B00
MCR>B00 [1,54]DSCSYS
DSC>DL0:/VEC=160
DSC>DL1:/VE=DL0:
```

Im folgenden Abschnitt einige Erlaeuterungen zur Systemgenerierung :

**MAPPED SYSTEM :** Der Adressierungsbereich ist groesser als 32K Worte. Ueber die Memory-Management-Unit (KT-11) werden 128K Worte adressiert.

**CHECKPOINTING :** Das betriebssystem verwaltet nach Prioritaeten den Ablauf mehrerer Programme in den gleichen Partitions.

**CONNECT TO INTERRUPT :** Es soll die Moeglichkeit gegeben werden, zu einem spaeteren Zeitpunkt Geraete ueber einen vom System angebotenen Interrupt- Vector mit einer HANDLER Task zu verbinden. Die hoechste Interruptadresse wurde mit 774 angegeben.

**USER ORIENTED TERMINAL DRIVER :** Dieser Terminal- Driver wurde gewaehlt, weil er den meisten zu erwartenden Anspruechen aller Anwendungen entspricht. Der Terminal- Driver wurde modifiziert, um das HP 2647 A Terminal mit einigen Zusatzfunktionen zu integrieren. (Kapitel 7)

**LINE FREQUENCY 50 HERTZ :** Das System bietet die Moeglichkeit relativ genaue Zeitpunkte zu markieren. Das minimal zu programmierende Zeitintervall betraegt einen TIC, d.h. 20m sec. Zusaetzlich kann in Einheiten von Sekunden, Minuten und Stunden der Rechner zu Aktivitaeten angestossen werden.

**RL11 DISC CONTROLLER :** Der Device-Driver fuer zwei RL11 Plattenlaufwerke wurde integriert.

**DL11 LINE INTERFACES :** Als Interfaces fuer die Ein- Ausgabe- terminals wurden zwei DL11 Line-Interfaces integriert.

1 DEC WRITER	120 Zeichen pro Zeile. 300 BAUD
1 HP 2647 A	80 Zeichen pro Zeile. 9600 BAUD
	- System Konsole -

Aufteilung des Memory :

PAR Ausdruck :

```
>PAR
LDR      000000 000000 MAIN TASK
FCP      120000 030000 MAIN TASK
MCR      150000 010000 MAIN TASK
GEN      160000 200000 MAIN SYS
          160000 010000 SUB  (...SYS)
USER     360000 270000 MAIN SYS
          360000 002600 SUB  (CAMA  )
          366600 002000 SUB  (INT24 )
INT1     650000 004000 MAIN SYS
          650000 002000 SUB  (DAH1  )
INT2     654000 004000 MAIN SYS
          654000 002000 SUB  (DAH2  )
EXCOM1   660000 020000 MAIN COM
EXCOM2   700000 020000 MAIN COM
EXCOM3   720000 020000 MAIN COM
EXCOM4   740000 020000 MAIN COM
EXPAGE   760000 020000 MAIN DEV
```

FCP-	Partition :	Adresse	20K.	Laenge	6K	(Worte)
MCR-	Partition :	Adresse	26K.	Laenge	2K	"
GEN-	Partition :	Adresse	28K.	Laenge	32K	"
USER	Partition :	Adresse	60K.	Laenge	46K	"
INT1	Partition :	Adresse	106K.	Laenge	1K	"
INT2	Partition :	Adresse	107K.	Laenge	1K	"
EXCOM1	Partition :	Adresse	108K.	Laenge	4K	"
EXCOM2	Partition :	Adresse	112K.	Laenge	4K	"
EXCOM3	Partition :	Adresse	116K.	Laenge	4K	"
EXCOM4	Partition :	Adresse	120K.	Laenge	4K	"
EXPAGE	Partition :	Adresse	124K.	Laenge	4K	"

Die GEN- Partition ist fuer die Utility- Programme vorgesehen.  
(d.h. Programmentwicklung)

In der USER- Partition sollen die Experimentprogramme laufen.  
(ABEA - TRIT - Initialisierungs- Auswerte- Protokollprogramme)

INT1 bietet der Interrupt-Routine I Platz. (ABEA- Anlage)  
INT2 bietet der Interrupt-Routine II Platz. (TRIT- Anlage)

EXCOM1 : Daten COMMON fuer ABEA  
EXCOM2 : Daten COMMON fuer TRIT  
EXCOM3 : Daten COMMON fuer die Abbildung aller Zustaende.  
EXCOM4 : Daten COMMON fuer die Auswertung.

EXPAGE : Reserviert fuer CAMAC- Adressen

## 6.2 BOOT - Vorgang und STARTUP - Aktivitaeten

In Ausnahmesituationen kann der BOOT- Vorgang erforderlich werden. Hierzu bedient man das Prozessor-Switch-Board :

a) CTRL und HALT

b) CTRL und BOOT

Der Rechner meldet sich auf der Systemkonsole (HP-2647/A) mit

000000 000024 165212 165530

@

und wartet auf die Eingabe des SYSTEM- Device. In diesem Fall

@DL (RETURN-Taste)

Das Betriebssystem wird von der DL0: ins Memory geladen und startet automatisch den STARTUP FILE. Der STARTUP FILE ist unter DL0:[1,2]STARTUP.CMD abgelegt und kann mit Hilfe des EDITOR- Programm dem jeweiligen Prozess angepasst werden.

Der unten aufgelistete STARTUP FILE ist in drei Funktionsgruppen aufgeteilt :

Teil I :

\*\*\*\*\*

- a) MOU DL1:ABEA
- b) SET /CRT=TT:
- c) SET /BUF=TT1:120.
- d) SET /BUF=TT0:120.
- e) ACS DL:/BLKS=1200.

- a) Die DL1: wird als Arbeitsplatte ins System integriert.
- b) TT0: ist ein Display- Terminal
- c) TT0: hat eine Schreibbreite von 120 Zeichen (Achtung LST !)
- d) TT1: hat eine Schreibbreite von 120 Zeichen (DEC-WRITER)
- e) Zuweisung von 1200 Bloecken auf DL0: fuer CHECKPOINTING

Teil II :

\*\*\*\*\*

```
INS [1,54]FOR /INC=20000
INS [1,1]EXPAGE/PAR=EXPAGE
INS [1,54]CAMA/PRI=200/PAR=USER
INS [1,54]ICO1/PRI=200/PAR=USER/TASK=...IC1
INS [1,54]ICO2/PRI=200/PAR=USER/TASK=...IC2
INS [1,54]ICO3/PRI=200/PAR=USER/TASK=...IC3
INS [1,54]ICO4/PRI=200/PAR=USER/TASK=...IC4
INS [1,54]VEC1/PRI=220/PAR=USER
INS [1,54]VEC2/PRI=220/PAR=USER
INS [1,54]VEC3/PRI=220/PAR=USER
INS [1,54]COP1/PAR=USER/TASK=...CP1
INS [1,54]COP2/PAR=USER/TASK=...CP2
INS [1,54]COP3/PAR=USER/TASK=...CP3
INS [1,54]COP4/PAR=USER/TASK=...CP4
```

```
INS [1,54]DAH1/PRI=200/PAR=INT1
INS [1,54]DAH2/PRI=200/PAR=INT2
INS [1,54]INT3/PRI=200/PAR=USER
INS [1,54]INT4/PRI=200/PAR=USER
INS [1,54]INT24/PRI=200/PAR=USER
INS [1,54]INVE/PAR=USER/TASK=...INV
INS [1,54]INCA/PAR=USER/TASK=...INC
INS [1,54]INMO/PAR=USER/TASK=...INM
INS [1,54]BEFL/PAR=USER/TASK=...BEF
INS [1,54]VENT/PAR=USER/TASK=...VEN
INS [1,54]LPHP/PAR=USER/TASK=...LST
INS [1,54]TEMU/PAR=USER/TASK=...TMU
INS [1,54]UHVI/PAR=USER/TASK=...UHV
INS [1,54]TEX1/PAR=USER/TASK=...EX1
INS [1,54]TEX2/PAR=USER/TASK=...EX2
INS [1,54]KALI/PAR=USER/TASK=...KAL
INS [1,54]EDON/PAR=USER/TASK=...EDO
INS [1,54]EDOF/PAR=USER/TASK=...EDE
```

#### Die INS > Installations-Anweisungen

Bei einer Installationsanweisung wird einer TASK eine feste Partition, eine vereinbarte Prioritaet und ein symbolischer Name wie z.B. ...LST zugewiesen. Letztere Namenszuweisung ermoeeglicht ein einfaches Aufrufen der TASK mit LST.

#### TEIL III \*\*\*\*\*

- a) REA ...LST 1 DL1:
- b) INC  
    RUN CAMA  
    INM  
    IC3  
    INV  
    RUN DAH1  
    RUN DAH2  
    RUN INT3  
    RUN INT4  
    RUN INT24
- c) BYE

zu a) Zuweisung der logischen Geraetennummer 1 auf DL1: (LST)  
zu b) Aufrufe von Initialisierungsprogrammen, den Interrupt-Routinen und des CAMAC - Handlers.

zu c) Mit dem BYE Kommando ist der zur Zeit einzige Benutzer aus dem SYSTEM ausgetragen. Ein neuer Zugriff zum Rechner kann nur nach vorrausgehendem "einloggen" erfolgen.

HEL [N,M]

Eingabe des PASSWORTES (falls bekannt)



## 7 S O F T W A R E \*\*\*\*\*

### 7.1 C A M A C - Software

Das gesamte CAMAC Software-Paket wurde nach den Gesichtspunkten einfacher Durchschaubarkeit, flexibler Gestaltung und modularer Technik aufgebaut. Dabei entwickelte sich von Ausbau- zu Ausbaustufe ein immer komfortableres Programmierungssystem.

Waehrend man in der ersten Ausbaustufe ein CAMAC-Modul mit seiner Adresse, Subadresse, CAMAC-Funktion, Bitweite, Synchronisierungs-Flag und eventuell mehr Angaben ansprechen musste, genuegen bei der letzten Ausbaustufe ausschliesslich von FORTRAN Programmen abgesetzte CALLS mit symbolischer Funktionsaussage.

Ein Beispiel fuer den selben CAMAC Aufruf in der tiefsten und hoechsten Programmiererebene.  
Lesen eines bestimmten ADC's :

#### a) Tiefste Ebene \*\*\*\*\*

```
CALL CAMAC (IRWC,ICFU,ILSD,IMOD,ICOM,IFLA,IDAT,IEXP,NADC)
```

Parameter :	IRWC	> READ, WRITE, oder CNTR
	ICFU	> CAMAC Funktion
	ILSD	> 24 oder 16 bit Datentransfer
	IMOD	> Modulnummer im CRATE (Stationsnummer)
	ICOM	> Anzusprechender COMMON- Bereich
	IFLA	> Synchronisations- Flag
	IDAT	> Daten (1 bzw. 2 Worte)
	IEXP	> Experimentbezeichnung
	NADC	> Nummer eines ADC's

#### b) Hoechste Ebene \*\*\*\*\*

```
CALL ADCN (NUAD, EXPE)
```

Parameter :	NUAD	> Nummer eines ADC's
	EXPE	> Experimentbezeichnung

### 7.1.1 Erstellung der External Page Software \*\*\*\*\*

Die Programmierung der CAMAC Hardware geschieht ueber die External Page. Von 124K bis 128K hat das System einen Adressbereich fuer die CAMAC Hardware-Adressen vorgesehen. Dieser Bereich geht als normales Memory verloren. Man kann sagen, CAMAC wird wie ein spezieller Memory-Bereich behandelt.

Bei der Systemgenerierung wurde eine Partition fuer die External Page mit folgender Anweisung angelegt:

```
SET /MAIN=EXPAGE:7600:200:DEV  
*****
```

Adressbereich 124K bis 128K .Das Partition Attribut DEV legt diese Partition als DEVICE- Partition aus.

#### External - Page - Assembler - Programm

```
.CSECT EXPAGE  
.BLKB 17700  
.END
```

Dieses Programm wird mit dem MACRO 11 ASSEMBLER uebersetzt.

```
MCR>MAC EXPAGE,EXPAGE=EXPAGE  
*****
```

Der OBJECT-FILE wird vom TKB wie folgt verarbeitet :

```
MCR>TKB  
TKB>DL:EXPAGE/PI, TI:,DL:EXPAGE/-HD=[3,1]EXPAGE  
TKB>/  
ENTER OPTIONS:  
TKB>PAR=EXPAGE  
TKB>STACK=0  
TKB>//
```

Die Expage-Task muss wie folgt installiert werden :

```
INS DL:[1,1]EXPAGE/PAR=EXPAGE  
*****
```

Die Expage Task kann wie ein COMMON-Bereich aufgefasst werden, den nur privilegierte TASKS lesen und schreiben koennen.

Anwendungsbeispiele siehe Kapitel 11 (Beispiele)

## 7.2 Beschreibung der CAMAC TASK

Die CAMAC Task laesst sich in fuenf Funktionsblocks einteilen:

I.

**WARTE  
STATUS**

II.

**DEKODIERUNG**

III.

**CAMAC ZUGRIFF**  
UEBER EXTERNAL PAGE

IV.

**CAMAC DATEN IN**  
COMMON BEREICHE

V.

**SYNCHRONISIERUNG**  
UEBER SYSTEM **FLAG**

Auf den folgenden Seiten sind die fuenf Blocks als Listing mit einigen Erlaeuterungen abgebildet.

TEIL I

```

;      CAMA STELLT DIE ALLEINIGE ZENTRALE TASK DAR, DIE
;      UEBER DIE EXTERNAL PAGE AUF DIE CAMAC HARDWARE
;      ZUGREIFT.
;
;      FROM SYSTEM: RBUF (1) = NAME DER SEND-TASK 1.
;                  RBUF (2) = NAME DER SEND-TASK 2.
;
;      VERSORGUNG:
;      *****
;      0, 1, 2      WBUF (1) = READ, WRITE, CNTR CODE
;                  WBUF (2) = CAMAC FUNCTION
;      0, 1         WBUF (3) = 16 BZW. 24 BIT TRANSFER
;                  WBUF (4) = CAMAC MODUL ADRESSE
;                  WBUF (5) = DATEN-BUFFER - READ -
;                  WBUF (6) = Q-TEST CODE
;                  WBUF (7) = FEHLER-FLAG
;                  WBUF (8) = SYNC-FLAG
;                  WBUF (9) = INTEGER HW
;                  WBUF (10) = INTEGER LW
;                  WBUF (11) = VORLAEUFIG FREI
;                  WBUF (12) = VORLAEUFIG FREI
;                  WBUF (13) = VORLAEUFIG FREI
;
;

```

.TITLE CAMA ZENTRUM

```

.MCALL SDAT$, SPND$, RCVD$, SETF$, DECL$
.MCALL EXIT$, RSUM$, DIR$, WTSE$, CLEF$
.MCALL QIO%C, CLEF$, SETF$

```

.NLIST BEX

CAMA:

SPND\$S

```

EMPF:  RCVD$S  ,#RBUF
       TST     $DSW
       BLT     CAMA

```

Die Task suspendiert sich. (SPND\$S). Das Betriebssystem aktiviert die Task, sobald ein anderes Programm Daten an diese schickt. Der Befehl RCVD\$S bewirkt die Uebernahme des SEND-Datenpuffers. Waehrend der Laufzeit der CAMAC Task werden alle Programme, die weitere Anforderungen an die CAMAC Task haben vom RSX 11M in eine Warteschlange eingereiht. Sobald Zeit vorhanden, arbeitet die CAMAC Task diese wartenden Programme mit den letzten drei Befehlen dieses Blocks ab.

TEIL II

```

;      *****
;      ADDRESS - D E C O D E R
;      *****

      MOV      WBUF      ,R0      ;READ, WRITE, CNTR
      ASL      R0
      ASL      R0
      TST      WBUF+4
      BEQ      L1
      INC      R0
      INC      R0

L1:      MOV      WBUF+6 ,R1      ;WELCHES CAMAC MODUL
      DEC      R1
      ASL      R1
      MOV      ADTA(R0) ,R2
      MOV      MOAD(R1) ,R3
      CMP      R0      ,#10
      BNE      L2
      BIC      #177740 ,WBUF+2
      BR L3

L2:      BIC      #177770 ,WBUF+2
L3:      MOV      R3      ,R4
      BIC      #1777      ,R4
      ADD      #1400      ,R4

      TST      R0
      BEQ      MG1
      CMP      R0      ,#2
      BEQ      MG2
      CMP      R0      ,#4
      BEQ      MG3
      CMP      R0      ,#6
      BEQ      MG4
      CMP      R0      ,#10
      BEQ      MG5

MG1:      MOV      R4      ,4(R2)
      MOV      R3      ,10(R2)
      MOV      WBUF+2 ,6(R2)
      BR      ALSPR

MG2:      MOV      R4      ,4(R2)
      MOV      R3      ,12(R2)
      MOV      WBUF+2 ,6(R2)
      BR      ALSPR

MG3:      MOV      R4      ,4(R2)
      MOV      R3      ,12(R2)
      MOV      WBUF+2 ,6(R2)
      BR      ALSPR

MG4:      MOV      R4      ,4(R2)
      MOV      R3      ,12(R2)
      MOV      WBUF+2 ,10(R2)
      BR      ALSPR

```

```

MGS:  MOV      R4      ,4(R2)
      MOV      R3      ,6(R2)
      MOV      WBUF+2  ,10(R2)
      BR       ALSPR

```

```

ALSPR: JMP      @ADTA(R0)      ; >>> READ, WRITE, CNTR

```

# Erlaeuterungen zum TEIL II

\*\*\*\*\*

In diesem zweiten Teil werden saemtliche CAMAC Parameter aus dem Empfaengerpuffer (RBUF: .BLKW 15. s.u.) uebernommen und dekodiert. Neben der CAMAC Modul Adresse wird der auszufuehrende CAMAC Code entschluesselt (READ, WRITE oder CNTR), die CAMAC Funktion und die Bitbreite des Datentransfers festgestellt. Weiterhin versorgt dieser Programmabschnitt den Teil III mit den verlangten Unterprogrammparametern.

## TEIL III

```

; *****
; C A M A C FUNCTION BLOCKS
; *****

```

R16Q:

```

JSR      R5      ,RQ
.WORD    A
.WORD    F
.WORD    A
.WORD    WBUF+20
.WORD    PUCO

JMP      PUCO

```

R24Q:

```

JSR      R5      ,RMQ
.WORD    A
.WORD    F
.WORD    WBUF+22
.WORD    A
.WORD    PUCO

JMP      PUCO

```

W16Q:

```

JSR      R5      ,WQ
.WORD    A
.WORD    F
.WORD    WBUF+20
.WORD    A
.WORD    SYNC

JMP      SYNC

```

```
W24Q:      JSR      R5      ,WMQ
           .WORD    A
           .WORD    WBUF+20
           .WORD    F
           .WORD    A
           .WORD    SYNC

           JMP      SYNC

COQ:       JSR      R5      ,CQ
           .WORD    A      ,&176000+1400
           .WORD    A
           .WORD    F      ,&37
           .WORD    SYNC

           JMP      SYNC
```

Dieser Programmabschnitt enthaelt die Aufrufe der Unterprogramme, die ihrerseits mit Hilfe direkter Adressierung der CAMAC- Geraete ueber die EXTERNAL PAGE den Datenaustausch zwischen CAMAC Task und CAMAC Hardware durchfuehren.

#### TEIL IV

```
PUCO:      MOV      WBUF+10 ,R0
           DEC      R0
           ASL      R0
           ASL      R0
           MOV      FOCO(R0) ,SENDE+2
           MOV      FOCO(R0) ,ACTI+2
           TST      (R0)+
           MOV      FOCO(R0) ,SENDE+4
           MOV      FOCO(R0) ,ACTI+4
           INC      WBUF+16
           MOV      WBUF+16 ,WAFL+2
           MOV      WBUF+16 ,CLFL+2
```

```
;          *****
;          ACTIVIERE COMMON TASK
;          *****
```

```
WRCO:      DIR$     #SENDE
           DIR$     #ACTI
           DIR$     #WAFL
           DIR$     #CLFL

           DEC      WBUF+16
           JMP      SYNC
```



Im vierten Teil der CAMAC Task werden die aus CAMAC gelesenen Werte im geforderten COMMON abgelegt. Hierzu wird mit einem SEND Aufruf pro COMMON die entsprechende Verwaltungs-Task mit Uebergabe der Parameter wie CAMAC Daten und Synchronisierungs-FLAG aufgerufen. Die CAMAC Task wartet auf die Fertigmeldung der aktivierten Verwaltungs-Task.

#### TEIL V

```
SYNC:  MOV      WBUF+16 ,FLAG+2  
        DIR$     #FLAG  
        DECL$S  
        JMP      EMPF
```

Der letzte Teil synchronisiert die CAMAC Task mit den aufrufenden Tasks. (EVENT-FLAG Mechanismus)  
Umseitig sind die abschliessenden Parameter und Tabellenfelder abgedruckt.

PARAMETER- und TABELLEN FELDER der CAMAC TASK  
\*\*\*\*\*

```
; *****
; DATEN-FELDER
; *****

RBUF:  .BLKW   15.
       WBUF = RBUF+4

A:      .WORD   0           ; DUMMY CAMAC ADR
F:      .WORD   0           ; DUMMY CAMAC FUNCTION

ADTA:   .WORD   R16Q        ; EINSPRUNG-TABELLE
       .WORD   R24Q
       .WORD   W16Q
       .WORD   W24Q
       .WORD   COQ
       .WORD   0

; *****
; SYSTEM-AUFRUFE MIT FREIEN PARAMETERN
; *****

SENDE:  SDAT$   FOCO,WBUF
ACTI:   RSUM$   FOCO
FLAG:   SETF$   FLAG
WAFL:   WTSE$   WAFL
CLFL:   CLEF$   CLFL

; *****
; C O M M O N - TASK TABELLE
; *****

FOCO:   .RAD50   /VEC1/
       .RAD50   /VEC2/
       .RAD50   /VEC3/
       .RAD50   /VEC4/
       .RAD50   /VEC6/
       .RAD50   /VEC6/
       .RAD50   /VEC7/
       .RAD50   /VEC8/
       .WORD    0
```

```

; *****
; C A M A C - ADRESSEN
; *****

MOAD: .WORD 166040 ;BORER D.W. DISPLAY (1)
      .WORD 166100 ;TELEFUNKEN PIR I (2) ABEA
      .WORD 166140 ;TELEFUNKEN POR I (3) ABEA
      .WORD 166200 ;KINETIC D A C 1 (4) ABEA
      .WORD 166202 ; D A C 2 (5) ABEA
      .WORD 166204 ; D A C 3 (6) ABEA
      .WORD 166206 ; D A C 4 (7) ABEA
      .WORD 166210 ; D A C 5 (8) ABEA
      .WORD 166212 ; D A C 6 (9) TRIT
      .WORD 166214 ; D A C 7 (10) TRIT
      .WORD 166216 ;KINETIC D A C 8 (11) TRIT
      .WORD 166240 ;TELEFUNKEN PIR II (12) TRIT
      .WORD 166300 ;TELEFUNKEN POR II (13) FREI
      .WORD 166340 ;JOERGER ADC I (14 -30) FREI
      .BLKW 16 ;PLATZ FUER 16 ADC'S FREI
      .WORD 166400 ;JOERGER ADC II 1 (31) ABEA
      .WORD 166402 ; ADC II 2 (32) ABEA
      .WORD 166404 ; ADC II 3 (33) ABEA
      .WORD 166406 ; ADC II 4 (34) ABEA
      .WORD 166410 ; ADC II 5 (35) ABEA
      .WORD 166412 ; ADC II 6 (36) ABEA
      .WORD 166414 ; ADC II 7 (37) ABEA
      .WORD 166416 ; ADC II 8 (38) ABEA
      .WORD 166420 ; ADC II 9 (39) ABEA
      .WORD 166422 ; ADC II 10 (40) ABEA
      .WORD 166424 ; ADC II 11 (41) TRIT
      .WORD 166426 ; ADC II 12 (42) TRIT
      .WORD 166430 ; ADC II 13 (43) TRIT
      .WORD 166432 ; ADC II 14 (44) TRIT
      .WORD 166434 ; ADC II 15 (45) TRIT
      .WORD 166436 ; ADC II 16 (46) TRIT

      .END CAMA

```

Durch Erweiterung oder Modifizierung kann gaenzlich andere CAMAC Hardware angesprochen werden.

Ueber die External Page lassen sich zur Zeit zwei (drei) CAMAC CRATES ansprechen.

Die maximale Datenrate ueber die CAMAC Task betraegt ca. 50 Zyklen pro Sekunde. (Memory - CAMAC Hardware)

Anmerkung:

Wie spaeter beschrieben, lassen sich Interrupts von CAMAC sehr leicht verarbeiten und bieten auf die Art der Interruptverarbeitung eine wesentlich bessere Zykluszeit.

(Die Zykluszeit ist abhaengig von der Laenge der INTERRUPT-SERVICE ROUTINE)

### 7.3 C A M A C INTERRUPT - ROUTINEN

Bei der Systemgenerierung wurde die CONNECT TO INTERRUPT Moeglichkeit eingebaut. Somit ist jedem CAMAC Modul ein Hardware-Interrupt-Vektor zugeordnet. Tritt ein Ereignis an einem Modul auf, wird ein diesem Vektor zugeordnetes Programm angestossen. Es wurden zwei Interrupt-Service-Routinen (ISR) implementiert. Die Vektoren 470 und 454, die den beiden TELEFUNKEN Parallel-Input-Registern zugeordnet sind, werden mit den entsprechenden ISR'n DAH1 und DAH2 verbunden.

7.3.1 Zur Erlaeuterung einer ISR soll das Beispiel der DAH1 ISR dienen.

TEIL I  
\*\*\*\*\*

```
;*****
;
;      PROGRAMMNAME           :      DAH1.MAC
;
;      BEARBEITER             :      H. HEER
;
;      ERSTELLUNG              :      06-DEC-79
;
;      LETZTE AENDERUNG       :      25-FEB-80
;
;*****
;
;      DAH1 BEARBEITET DIE INTERRUPTS VOM DANA ANS PIR1
;      DIE INTERRUPT DATEN WERDEN IN EXCOM3 ABGELEGT- KOPA1 (27-28)
;      MESSWERTERFASSUNG DER  A B E A  = ANLAGE  (TEMPERATUR)
;
;      .MCALL CINT$,EXIT$$, QIOW$, DIR$, WTSE$$, CLEF$$

      LUN.TT = 5
      EFN.TT = 2
      EFN.WF = 1

$DAH1:  DIR$      #HELLO           ;MSG: INTER. CONNECTED
        DIR$      #CINT           ;CONNECT TO INTERRUPT
        BCS        ERR1
WARTE:  WTSE$$    #EFN.WF ;WAIT FOR DANA BUFFER COMPLETE

        CLEF$$    #EFN.WF ;CLEAR FLAG

        MOV        BUFF      ,ABEA+84.
        MOV        BUFF+2    ,ABEA+86.

        JMP        WARTE

        DIR$      #DISCON
        DIR$      #BYE
        EXIT$$
```

```
ERR1:  MOV      #1      ,R0      ;ERROR #1
        MOV      $DSW    ,R1
        IOT

HELLO:  QIOW$     IO.WLB,LUN.TT,EFN.TT,,,,<BU1,LA1>
BYE:    QIOW$     IO.WLB,LUN.TT,EFN.TT,,,,<BU2,LA2>

CINT:   CINT$     $VECTR,$BASE,$DAISR,$DAEDI,PR4
DISCON: CINT$     $VECTR,0,0

        .NLIST    BEX
BU1:    .ASCII    <15><12>/A B E A  DANA-INPUT CONNECTED/
        LA1=.-BU1
BU2:    .ASCII    <15><12>/A B E A  DANA-INPUT DISCONNECTED/
        LA2=.-BU2
        .EVEN
        .LIST     BEX

        .PSECT    EXP3,GBL,OVR,D
ABEA:   .BLKW     80.
        .CSECT
```

Im ersten Teil des Programms wird die Verbindung zwischen Hardware-Interrupt und ISR hergestellt. Ausserdem sind Meldungen ueber erfolgreiche bzw. missglueckte Interruptverbindungen vorgesehen. Sobald die ISR angebunden ist, geht dieser erste Teil in einen Wartestatus. Er wartet auf eine Local Flag, die bei Auftreten eines Interrupts von der eigentlichen ISR gesetzt wird. Anschliessend speichert Teil I den aus dem PIR ausgelesenen Wert im Daten-COMMON EXP3 ab und geht wieder in den Wartestatus.

TEIL II  
\*\*\*\*\*

```
;*****
$BASE:                                     ;BASE ADDRESS
;*****

      CSR = 167400
      PIR = 166100

TSKTCB: .WORD    0          ;TCB ADDRESS OF TASK

;      ENABLE / DISABLE ROUTINE
;      *****

$DAEDI: BCS      20$
        BIS      #40      ,@#CSR
        MOV      @#$TKTCB,TSKTCB ;COPY TASK TCB ADR

        RETURN

20$:    BIC      #40      ,@#CSR

        RETURN
```

Der Teil II bildet die Befehle fuer die ENABLE bzw. DISABLE Interrupt von CAMAC Anweisungen. Beim Durchlauf der ENABLE Interrupt Routine wird gleichzeitig der eigentlichen ISR die Informationen uebergeben, die fuer das Setzen der LOCAL FLAG erforderlich sind.

TEIL III  
\*\*\*\*\*

```
;*****
;      INTERRUPT SERVICE ROUTINE
;*****

$DAISR:
;      *****
;      SAVE REGISTER
;      *****
```

```

MOV      R0      ,-(SP)
MOV      R1      ,-(SP)
MOV      R2      ,-(SP)
MOV      R3      ,-(SP)
MOV      R4      ,-(SP)
MOV      R5      ,-(SP)

MOV      #CSR     ,R0
BIC      #7       ,(R0)
BIS      #2       ,(R0)+
MOV      (R0)     ,R3          ;SAVE HIGH MANTISSA
MOV      @#PIR    ,R1
MOV      (R0)     ,R2
MOV      R3       ,(R0)      ;UNSAVE HIGH MANTISSA

COM      R1
COM      R2

MOV      R1       ,BUFF+2
MOV      R2       ,BUFF

MOV      #EFN.WF  ,R0      ;GET EF.N TO SET
MOV      TSKTCB   ,R5      ;GET TASK TCB-ADR.

CALL     @##SETF          ;SET THE LOCAL EV.FL.

;      *****
;      UNSAVE REGISTER
;      *****

MOV      (SP)+    ,R5
MOV      (SP)+    ,R4
MOV      (SP)+    ,R3
MOV      (SP)+    ,R2
MOV      (SP)+    ,R1
MOV      (SP)+    ,R0

; *****
;      ENABLE DANA INTERRUPT
; *****

BIS      #40      ,@#CSR

RETURN

```

Teil III stellt die eigentliche ISR dar. Die Moduladresse 166100 und die CRATE-CONTROLLER- Adresse 167400 sind fest definiert. Wichtig ist, dass die alte HIGH MANTISSA d.h. die oberen 8 bit eines 24 bit CAMAC Wortes waehrend des Lesens des PIR temporaer gerettet werden. Nach dem Ablegen des Wertes in einem Programm-Hilfspuffer wird der naechste Interrupt zugelassen. Zuletzt setzt die ISR die LOCAL FLAG fuer den Teil I des DAH1 Programms.



#### 7.4 C A M A C Initialisierung

Bei der STARTUP Phase wird das Programm INCA gestartet.  
Es initialisiert das gesamte CAMAC System.  
Dieses Programm darf niemals waehrend eines laufenden Ex-  
periments aufgerufen werden!

(Clear CAMAC)

Programmlisting :

```
;*****  
;  
;      PROGRAMNAME           :      INCA.MAC  
;  
;      BEARBEITER            :      H. HEER, P. MECKING  
;  
;      ERSTELLUNG            :      19-FEB-80  
;  
;      LETZTE AENDERUNG      :      19-FEB-80  
;  
;*****  
;  
;      INCA INITIALISIERT DAS GESAMTE C A M A C SYSTEM  
;  
;      ADRESS - DEFINITIONEN  
;      *****  
;      CIN = 166010  
;      CCC = 167400  
;  
;      .MCALL EXIT%C  
  
START:  CTRL    1. 0,CIN 0  
        BIC     #10      ,CCC  
        EXIT%C  
        .END     START
```

Ueber die Adresse CIN = 166010 wird ein Clear CAMAC Zyklus  
eingeleitet. Zweite Operation ist das Ruecksetzen des INHIBIT  
TO THE CAMAC BUS Bit im CONTROL STATUS REGISTER des CAMAC  
CRATE CONTROLLERS.

## 7.5 Initialisiere CAMAC Module

Bei der Startup Phase wird das Programm INM0 gestartet.  
Es dient zur Vorbereitung einzelner CAMAC Module.

Beispiele:

Modul einschalten  
Modul Status einschreiben  
u.s.w.

INM0 Programmlisting :

```
C
C      INMO SCHALTET C A M A C - MODULE EIN
C
C      INTEGER IDAT (2)
C
C      CALL ALFL (IFLA)
C
C      *****
C      POR - TELEFUNKEN OUTPUT-REG. I UND II EIN
C      *****
C
C      CALL CAMAC (2,26,0,3,IFLA)
C      CALL CAMAC (2,26,0,13,IFLA)
C
C      *****
C      PIR - TELEFUNKEN INPUT-REG. I UND II EIN
C      *****
C
C      CALL CAMAC (2,26,0,2,IFLA)
C      CALL CAMAC (2,26,0,12,IFLA)
C
C      *****
C      A D C INITIALISIEREN ( SCAN 16 )
C      *****
C
C      IDAT (2) = 0
C
C      CALL CAMAC (1,16,0,31,0,IFLA,IDAT)
C
C      CALL FRFL (IFLA)
C
C      CALL EXIT
C
C      END
```

## 8 Software zum HP 2647/A

Die spezielle Hardwarekonfiguration PDP 11/34, HP 2647/A und der Graphische Printer HP 2631/G, der ueber das IB Interface am HP 2647/A angeschlossen ist, erforderten einige Programme, die Schritt fuer Schritt eine Anpassung des Printer-Plotters als Schnelldrucker an den Rechner ermoeeglichten. Dies liess sich durch die vom HP Terminal angebotenen Escape-Sequenzen loesen. Wir unterscheiden fuenf Arten von Software- Unterstuetzungen des HP Terminals :

- 8.1 Software auf dem HP 2647/A Terminal - OFF LINE -
- 8.2 PDP Software zur Vorbereitung des Terminals.
- 8.3 PDP Software zur Benutzung des 2631/G als Drucker
- 8.4 PDP Software zur Benutzung der HP Terminal Kassetten
- 8.5 Terminal BASIC (OFF LINE) aktiviert PDP 11 Software

Allgemeine Software speziell RSX 11M V3.1  
\*\*\*\*\*

Vorraussetzung fuer die gesamte Terminal Software ist der modifizierte Terminal Handler.

- a) Einbau der ENQUIRE - ACKNOWLEDGE Faehigkeit (Timing)
- b) Einbau der TIME OUT Routinen speziell fuer das 2647/A
- c) Erweiterung und Modifizierung des TER - Programm

zu c : Es kann ueber das TER Kommando gesetzt und abgefragt werden, welche Terminals HP 2647/A Typen sind. (RSX 11M V3.1)

### 8.1 HP Terminal OFF LINE - HP 2647/A BASIC

Es werden Messdaten mit Hilfe des CIP Programms (s.u.) auf eine HP Kasette geschrieben.  
Ein BASIC Programm uebernimmt die graphische Darstellung der Daten im Terminal Multi Plot Mode.  
Auch die Daten fuer das Menu- feld koennen auf einer Kasette vorbereitet sein.

## 8.2 HP 2647/A ON LINE

### 8.2.1 PDP Software zur Terminalunterstützung

Es wurde ein Unterprogramm entwickelt, das alle wichtigen Escape-Sequenzen ausführen kann. Das Programm wird mit einem Parameter, der Escape-Sequenz-Nummer aufgerufen.

Beispiel : CALL ESCA (10)  
Ausführung : CLEAR MEMORY des HP2647/A

Liste der bisher implementierten Escape-Sequenzen

\*\*\*\*\*

Nummer	Aktivitäten
--------	-------------

\*\*\*\*\*

C\*\*\*\*\*

C DEFINITIONEN DER ESCAPE - SEQUENZEN

C\*\*\*\*\*

C

C1	*****	CURSOR UP	*****
C11	*****	CLEAR LINE	*****
C12	*****	SCROLL DISPL.UP	*****
C13	*****	SCROLL DISPL.DO	*****
C14	*****	NEXT PAGE	*****
C15	*****	PREVIOUS PAGE	*****
C16	*****	START UNPRO. F.	*****
C17	*****	END UNPROT. F.	*****
C18	*****	FORMAT MODE ON	*****
C19	*****	FORMAT MODE OFF	*****
C20	*****	INSERT LINE	*****
C21	*****	DELETE LINE	*****
C22	*****	DELETE CHARAC.	*****
C23	*****	DEL.CHAR.WR.AR.	*****
C24	*****	INSERT CHA.ON	*****
C25	*****	INSERT CHA.OFF	*****
C26	*****	IN.CHA.WR.AR.ON	*****
C27	*****	I.CHA.WR.AR.OFF	*****
C28	*****	CAPS LOCK ON	*****
C29	*****	CAPS LOCK OFF	*****

```
C30      ***** MEMORY LOCK ON *****
C31      ***** MEMORY LOCK OFF *****
C32      ***** AUT.LIN.FEED ON *****
C33      ***** AUT.LIN.FE. OFF *****
C34      ***** REMOTE ON *****
C35      ***** REMOTE OFF *****
C36      ***** BLOCK MODE ON *****
C37      ***** BLOCK MODE OFF *****
C38      ***** DISPL. FUNC. ON *****
C39      ***** DISP. FUN. OFF *****
C40      ***** RES. TERMI I *****
C41      ***** RES. TERMI II *****
C42      ***** TERMI.SELF TEST *****
C43      ***** DELAY 1 SECOND *****
C44      ***** KEYBOARD ENABLE *****
C45      ***** KEYBOARD DISABL *****
C46      ***** BL.TRA.ENA FR.C *****
C47      ***** DIPL.US.DF.SO-K *****
C48      ***** DI.U.D.SO-K OFF *****
C49      ***** TERMINAL STATUS *****
C50      ***** EXTENDED STATUS *****
C51      ***** ASS SOU > LE-TA *****
C52      ***** ASS DES > IB #1 *****
C53      ***** REWIN LEFT TAPE *****
C54      ***** FIND F 5 LEFT T *****
C55      ***** FIND F 4 LEFT T *****
C56      ***** B A S I C *****
C57      ***** REMOVE STD X *****
C58      ***** GET "L" *****
C59      ***** FIND F 1 LEFT T *****
C60      ***** R U N *****
C61      ***** DISPL.WO.SPAC 1 *****
C62      ***** PLOT A X E S *****
C63      ***** MULTI P L O T *****
C64      ***** DISPL.WO.SPAC 2 *****
C65      ***** LOAD LEFT TAPE *****
C66      ***** SET SIZE 10000 *****
C67      ***** PLOT A X E S *****
C68      ***** M U L T I P L O T *****
C69      ***** BASIC EXIT *****
C70      ***** ASS SOURCE > DA *****
C71      ***** ENABLE RECORD *****
C72      ***** DISABLE RECORD *****
C73      ***** ASS DESTI > DI *****
C74      ***** ASS SOURC > DI *****
C75      ***** C O P Y ALL *****
C76      ***** EDIT MODE ON *****
C77      ***** EDIT MODE OFF *****
C78      ***** SEND 120 DATA *****
C79      ***** TRAN F DA > DES *****
C*****
```

## 8.2.2 Vorbereitung des HP 2647/A fuer graphische Anwendungen

Die HP 2647/A Software wurde auf einer HP-Kassette geliefert. Auf dieser sind auf verschiedenen FILES die einzelnen Programme abgelegt, die nach Wunsch ueber Tastenbedienung am Terminal von der Kassette ins Memory des Terminals gelesen werden. Diese Handbedienung kann durch Benutzung der folgenden Programme umgangen werden.

### a) INMU - Initialisiere MULTI-PLOT

```
C*****
C
      DIMENSION MULT(11)
C
      MULT(1) = 7
      MULT(2) = 10
      MULT(3) = 55
      MULT(4) = 56
      MULT(5) = 57
      MULT(6) = 58
      MULT(7) = 59
      MULT(8) = 60
      MULT(9) = 7
      MULT(10)= 10
      MULT(11)= 61
C
      DO 10 I=1 , 11
C
      IF (I.NE.9) GO TO 10
C
      CALL MARK (1,20,2)
      CALL WAITFR (1)
C
10    CALL ESCA (MULT(I))
C
      CALL EXIT
```

### Erlaeuterung :

Das Laden des MULTI-PLOT erfordert den Aufruf von 11 Escape-Sequenzen. Die Nummern dieser Sequenzen werden in einem Feld abgelegt. Das Unterprogramm ESCA benutzt in einer Schleife aufgerufen als Parameter das Datenfeld mit den Escapenummern. Lediglich nach der RUN Anweisung, der Escapenummer 60 musste ins Programm ein Warte-Aufruf eingebaut werden. Das Programm wartet 20 Sekunden. Die HP Kassette muss im linken Kassetten-Laufwerk stecken.

b) BALO BASIC - LOAD - Routine

Die BASIC HP Kasette muss im linken Laufwerk stecken.  
Es werden nacheinander die 6 erforderlichen Escape-Sequenzen ans Terminal abgeschickt.

Programm - Listing

\*\*\*\*\*

```
C*****
C
C      BALO LAEDT BASIC VON DER LINKEN KASSETTE
C
C      INTEGER BAFE (6)
C
C          BAFE (1) = 54
C          BAFE (2) = 65
C          BAFE (3) = 59
C          BAFE (4) = 69
C          BAFE (5) = 7
C          BAFE (6) = 10
C
C      DO 10 I = 1 , 6
C
C10      CALL ESCA(BAFE(I))
C
C          CALL EXIT
C
C      END
```

c) LAME - Lade MENU - Feld

Dieses Beispiel zeigt die Datenkommunikation zwischen Rechner und HP 2647/A auf tiefster Programmiererebene. Es werden nicht nur die herkoemmlichen Escape-Sequenzen ans Terminal geschickt sondern auch vorbereitete Datenfelder. In diesem Fall enthalten die Datenfelder die Informationen fuer das MENU- Feld des Terminals fuer den spaeteren MULTI-PLOT Mode. Umseitiges Listing des LAME Programms zeigt unter anderem auch ein Beispiel des "QIO" Aufrufes, eine dem Assembler schon sehr nahe stehende Programmierungshilfe. Es wird unter anderem das Enquire - Acknowledge Handshake angestossen. Die im letzten Programmabschnitt aufgerufene TASK PUDA schickt die zu plottenden Daten ans Terminal. Das Terminal uebernimmt die Beschriftung und Scalierung der Zeichnung aus den Informationen des MENU- Feldes.



```
C*****
C
C      LADE MENU-FELD FUER MULTILOT
C
C      LOGICAL *1 INP, ACK, D(56)
C      INTEGER RPR, IPAR(6), MULT (8)
C
C      DATA RPR/"4432/
C      DATA ACK/"6/
C      DATA PUDA/6RPUDA /
C
C***** M E N U E - D A T E N *****
C      DATA D/"62,"11
C      3,"104,"101,"61,"11,"11,"62,"11,"61
C      3,"11,"11,"65,"70,"60,"11,"11,"11,"131
C      4,"11,"60,"11,"62,"65,"60,"11,"64,"60
C      4,"67,"60,"11
C      5,"61,"60,"60,"11,"60,"11,"61,"60,"11
C      5,"63,"60,"11,"61,"11
C      6,"11,"111,"116,"104,"11,"11
C      7,"126,"117,"114,"124,"5/
C*****
C
C      MULT (1) = 64
C      MULT (2) = 7
C      MULT (3) = 10
C      MULT (4) = 61
C      MULT (5) = 7
C      MULT (6) = 10
C      MULT (7) = 67
C      MULT (8) = 68
C
C      DO 10 I = 1 , 3
C
C      CALL ESCA (MULT(I))
C
C      CALL GETADR (IPAR(1),INP)
C      IPAR(2) = 1
C
C      CALL GETADR (IPAR(4),D)
C      IPAR(5) = 56
C
C      CALL WTQIO (RPR,6,3,,,IPAR,IDS)
C
C      DO 20 I = 4 , 8
C
C      CALL ESCA (MULT(I))
C
C      IF (I.LT.7) GO TO 20
C
C      CALL MARK (1,10,2)
C      CALL WAITFR (1)
C
C      CONTINUE
C
C      CALL REQUES (PUDA)
C
C      CALL EXIT
C
C      END
```

### 8.3 HP 2631/G als Schnelldrucker unter RSX 11M V3.1

Der Weg bis zur entgueltig brauchbaren Loesung fuehrte Ueber einige Ausbaustufen.

#### a) Benutzung der EDIT ON/OFF FUNCTION

Mit den Programmen EDON und EDOF laesst sich die EDIT Funktion am Terminal ein- bzw. ausschalten. Zusaetzlich werden Quellgeraet in diesem Fall das DATA COMM Interface als Nahtstelle der Rechnerdaten und als Zielgeraet das HP-IB Interface als Nahtstelle fuer die Schnelldruckerdaten festgelegt. Die vom Rechner geschickten Daten werden bis zum Ueberlauf ins Terminal Memory geladen. Beim Ueberlauf erscheinen die sog. uebergelaufenen Daten auf dem 2631/G. Nach Beendigung der Rechnerdaten koennen die restlichen noch im Terminal-Memory verbliebenen Daten durch das betaetigen der RECORD Taste auf den Drucker geschickt werden. Ein gangbarer, wenn auch etwas umstaendlicher Weg.  
Groesster Nachteil : Es werden von der angebotenen Schriftbreite (120) des Druckers nur 80 (Terminal) ausgenutzt.

#### b) Das LPHP Programm (Line Printer Hewlett Packard)

Listing des LPHP Programm.  
\*\*\*\*\*

```
C*****
C
C      LPHP SOLL ASCII - FILES AUF DIREKTEM WEGE UEBER
C      DEN HP - IB BUS AUF DEM PRINTER PLOTTER 2631G
C      DRUCKEN. (PRO ZEILE 120 ZEICHEN !). WAEHREND DER
C      AUSGABE IST DAS TERMINAL HP 2647A GESPERRT.
C      ERLAUBTE EXTENTIONS : LST, MAP, ASC
C
C      LOGICAL *1 IFEL (120), CR, FF, LF, PUNKT
C      LOGICAL *1 IFIN (120), INAM (10), LST (4)
C      LOGICAL *1 FOFI (4), NUL
C      INTEGER ISEQ (3), Q
C
C      DATA IFIN /"14,118*"40,"14/
C      DATA CR /"15/, FF /"14/, LF /"12/, PUNKT /"56/
C      DATA LST /"56,"114,"123,"124/, NUL /"0/
C      DATA INAM /10*"40/, FOFI /"33,"52,"162,"104/
C
C      CALL ERRSET (29,.TRUE.,.FALSE.,.TRUE.,.FALSE.)
C
C      DO 7 I = 1 , 10
C      INAM (I) = "40
7
```

```
CALL GONG
C
WRITE (6,70)
70  FORMAT ('%GIB FILE-NAME IN DER FORM XXXXXX.XXX : ')
C
READ (6,75,END=300) Q, (INAM(K), K=1,Q)
75  FORMAT (Q,10 A1)
C
IF ((Q .EQ. 0) .OR. (Q .GT. 10)) GO TO 300
C
IPUN = 0
C
DO 2 I = 1 , Q
2   IF (INAM(I) .EQ. PUNKT) IPUN = 1
C
IF (IPUN .EQ. 1) GO TO 3
C
IF (Q .GT. 6) GO TO 300
C
DO 4 I = 1 , 4
4   LAU = Q + I
   INAM (LAU) = LST (I)
C
Q = Q + 4
C
3   CALL ASSIGN (1,INAM,Q)
C
CALL FDBSET (1,'READONLY')
C
ISEQ(1) = 70
ISEQ(2) = 52
ISEQ(3) = 79
C
CALL CLEAR
C
CALL ESCA (45)
C
WRITE (6,110)
110  FORMAT (1H , 'TERMINAL WAEHREND DES DRUCKENS GESPERRT !'//)
C
DO 20 I = 1 , 3
20   CALL ESCA (ISEQ(I))
C
DO 10 I = 1 , 2000
C
DO 120 IL = 1 , Q
120  IFEL (IL) = "40
C
READ (1,40,END=100,ERR=100) Q, (IFEL(K),K=1,Q)
40  FORMAT (Q, 120 A1)
```

```

C      DO 50 L = 1,Q
C      ILA = L
C      IF (IFEL(L) .EQ. CR) GO TO 60
50     IF (IFEL(L) .EQ. NUL) GO TO 64
C
60     IF (ILA .NE. 1) GO TO 65
C
      KA = ILA + 1
C
      DO 80 ISU = KA, Q
      ILA = ISU
80     IF (IFEL(ISU) .EQ. CR) GO TO 65
      ILA = 2
      GO TO 65
C
64     IFEL(ILA) = CR
C
65     WRITE (6,30) (IFEL(K), K=1, ILA)
30     FORMAT (1H ,120 A1)
C
10     CONTINUE
C
100    WRITE (6,200) (FOFI(K), K=1,4)
C
      WRITE (6,200) (IFIN(K), K=1,120)
200    FORMAT (1H ,120 A1)
C
      CALL ESCA (44)
C
      DO 1 I = 1 , 8
1      CALL ESCA (1)
C
      WRITE (6,90) (INAM(L), L=1,10)
90     FORMAT (1H ,10 A1,' GELISTET. TERMINAL FREI !'//)
C
300    CALL EXIT
C
      END
```

Dieses Programm listet auf direktem Wege also ohne ueber den Terminal Display Datensaeetze des Rechners auf dem HP-Printer-Plotter im Format 120 Zeichen pro Zeile.  
Zur Zeit werden Files mit folgender Extension gelistet :

.LST .MAP .ASC .CMD

Ausbau beliebig moeglich. (.BIN .OBJ ...)  
Waehrend des Druckvorgangs ist das Terminal gesperrt.  
Vorraussetzung : Modifizierter TT-Driver des RSX 11M v3.1

#### 8.4 Software zur Unterstuetzung der HP Kassetten

Das Programm BIGCIP ist in der Lage, Files von jedem beliebigen RSX-Device zu den Cassetten oder dem Printer eines HP264X-Terminals zu transferieren bzw. umgekehrt. Das Band jeder Cassette muss manuell positioniert werden. Es wird immer von der jeweiligen Bandposition an gelesen bzw. beschrieben. Beim Schreiben kann man mit dem Switch /END das Ueberschreiben der bereits auf dem Band befindlichen Files verhindern, der neue File wird hinter den letzten File geschrieben. Jeder File wird auf der Cassette durch eine File-Marke beendet. Der Benutzer hat selbst dafuer zu sorgen, dass hinter den letzten File eine END-OF-DATA-Marke geschrieben wird.

Beim Lesen von der Cassette wird immer nur ein File uebertragen, dessen Ende durch eine File-Marke, die END-OF-VALID-DATA-Marke oder durch das Ende des Bandes bestimmt wird. Wird der Switch /ALL angegeben, so werden alle Files einer Cassette von der jeweiligen Bandposition an uebertragen. Beim Kopieren zu einem RSX-Device wird das Ende jedes Files durch einen Seitenvorschub ("FORM-FEED") angezeigt, beim Kopieren auf eine zweite Cassette wird eine File-Marke hinter jeden File geschrieben.

Die Files koennen in zwei Modes uebertragen werden. Der erste Mode ist Default und stellt einen ASCII-Transfer dar. Es werden alle "Display-Enhancement-Escape-Sequenzen" aus den Records entfernt sowie der "Format-Mode" entschluesselt.

Im anderen Falle wird durch den Switch /BIN ein Binary-Transfer veranlasst, der alle Records unveraendert laesst. Beim ASCII-Mode koennen Tab-Konvertierungen vorgenommen werden, sodass ein uebertragener Record keine Tabs, nur einen Tab oder ueberall wo moeglich Tabs enthaelt. Default ist der Mode, dass die Records bezueglich der Tabs unveraendert bleiben. Ansonsten ist einer der Switches /TAB:0 oder /TAB:1 oder /TAB:A anzugeben.

Alle Switches koennen sowohl bei der Input- als auch bei der Output-Specifikation angegeben werden. Ebenso sind einige Kombinationen der Switches moeglich.

Das Programm BIGCIP wird als "CIP" installiert und kann auf zwei Arten aufgerufen werden:

```
MCR>CIP "Command"  
>
```

oder

```
MCR>CIP  
CIP>"Command"  
CIP>...  
CIP>^Z
```

>

Jeder "Command" hat folgende allgemeine Form:

```
"FILE-SPEZIFIKATION"[/SW][SW]...="FILE-SPEZIFIKATION"[/SW][SW]...  
oder  
"FILE-SPEZIFIKATION"[/SW][SW]...=[TTNNN:](SUB)[/SW][SW]...  
oder  
[TTNNN:](SUB)[/SW][SW]...=[TTNNN:](SUB)[/SW][SW]...  
oder  
[TTNNN:](SUB)[/SW][SW]...="FILE-SPEZIFIKATION"[/SW][SW]...
```

Der Ausdruck "FILE-SPEZIFIKATION" stellt eine RSX11M-uebliche File- bzw. Device- und/oder UIC-Angabe dar.

[/SW] zeigt einen oder mehrere Switche an, die optional sind. Folgende Switche sind definiert:

/ALL	es werden alle Files einer Cassette von der augenblicklichen Position des Bandes an uebertragen.
/BIN	es wird ein File von einer Cassette binaer uebertragen, es werden weder die "Display-Enhancement-Escape-Sequenzen" aus den Records entfernt noch der "Format-Mode" entschluesst.
/END	der neue File wird hinter den letzten bestehenden File auf eine Cassette geschrieben.
/TAB:Ø	es werden alle Tabs aus den Records entfernt und durch Blanks ersetzt.
/TAB:1	es werden alle Tabs aus den Records entfernt. Dort wo moeglich wird in den ersten 8 Stellen eines Records ein Tab an Stelle von Blanks eingefuegt.
/TAB:A	ueberall wo moeglich werden Blanks eines Records durch Tabs ersetzt.

Die TAB-Switches duerfen nicht widerspruechlich benutzt werden, das heisst, es kann nur ein Tab-Switch pro "Command" angegeben werden. Der BIN-Switch kann nicht mit /ALL oder einem TAB-Switch kombiniert werden.

[TTNNN:] stellt eine RSX11M-uebliche Terminalangabe dar. Fehlt diese, so wird TI: als Default gesetzt.

(SUB) stellt eine Subline-Number des HP-264X-Terminals dar. Folgende Bezeichnungen sind fuer die einzelnen Substationen definiert:

(1) oder (L)	LEFT TAPE
(2) oder (R)	RIGHT TAPE
(3) oder (D)	DISPLAY
(4) oder (P)	PRINTER

Sollte aus unerklaerlichen Gruenden einmal das Programm BIGCIP in einen sogenannten "DEAD-LOCK"-Zustand geraten, so muss maximal 80 Sekunden gewartet werden, bis BIGCIP durch den TIMEOUT-Count wieder aktiviert wird.

#### 8.5 Graphische Darstellung mittels HP BASIC : Vom Rechner ermittelte und uebermittelte Daten

Zum Bedienungszeitpunkt ist das Terminal OFF LINE.  
Das BASIC Programm schaltet das Terminal ON LINE und startet im Rechner ein beliebiges Programm. Dieses Programm errechnet die Graphikdaten und sendet sie zum Terminal. Eine Handshake-Routine im BASIC Programm steuert die Uebername der Rechnerdaten. Zu den X und Y Koordinaten werden auch die Parameter fuer die Bildgroesse und die Skalierungsfaktoren ans Terminal gesendet. Vor dem Plotvorgang d.h. nach Uebernahme aller Rechnerdaten (Anzahl auch vom Rechner) schaltet das BASIC Programm wieder in den OFF LINE Zustand.

Umseitiges Programm veranschaulicht einen moeglichen Weg zur Kommunikation zwischen Rechneraktivitaeten und HP 2647/A BASIC Programmen.

```
100 REM *****
110 REM DEFINE CONTR. FUNCTIONS
120 Rmote$=CHR$(27)&"&k1R"
130 Loc1$=CHR$(27)&"&k0R"
140 DIM N1$(4)
150 DIM X(500)
160 DIM Y(500)
170 INPUT "GIB PROGRAMM-NAME (XXXX) : ",N1$
180 REM *****
190 REM PUT TERMINAL IN REMOTE STATUS
200 PRINT Rmote$;
210 REM *****
220 REM SET DATA-COMM TO LOG. UNIT 1
230 ASSIGN "DA" TO #1
240 REM *****
250 REM RUN CPU BASIC PROGRAM
260 PRINT #1;"RUN "&N1$&CHR$(13);
270 GOSUB 380
280 READ #1;N1,N2,S1,S2,S3,S4,X0,X1,Y0,Y1
290 READ #1;C1,C2
295 FOR L=1 TO N1\A1=(L-1)*10+1\E1=A1+9
300 FOR I=A1 TO E1\ READ #1;Y(I)\ NEXT I
301 NEXT L
302 FOR I=1 TO N2\ READ #1;Y(E1+I)\ NEXT I
305 FOR L=1 TO N1\A1=(L-1)*10+1\E1=A1+9
310 FOR I=A1 TO E1\ READ #1;X(I)\ NEXT I
315 NEXT L
316 FOR I=1 TO N2\ READ #1;X(E1+I)\ NEXT I
320 REM *****
330 REM PUT TERMINAL IN LOCAL STATUS
340 PRINT Loc1$;
350 GOTO 440
360 REM *****
370 REM DATA COMM HANDSHAKE ROUTINE
380 GETDCM ON \D$=""
390 IF GETDCM(D$)=0 OR D$(>)CHR$(13) THEN 390
400 GETDCM OFF
410 RETURN
420 REM *****
430 REM DRAW AND LABEL GRID ( A G L )
440 X2=X1/10
450 Y2=Y1/10
460 PLOTR
470 LOCATE (S1,S2,S3,S4)
480 SCALE (X0,X1,Y0,Y1)
490 FXD (C1,C2)
500 LGRID (X2,Y2,0,0,2,2)
510 LINE (1)
520 FOR I=1 TO N1*10+N2 STEP 1
530 PLOT (X(I),Y(I))
540 NEXT I
550 END
```



## 9        FORTRAN   C A M A C   CALLS

Bei der Entwicklung der Software wurde besonderer Wert auf leichte Bedienung, und einfache Handhabung der einzelnen Programmkomponenten gelegt. Speziell die CAMAC-Nahtstellen betreffend kamen wir auf sehr einfache Aufrufe. Somit steht zukünftigen Änderungen an der gesamten Prozess-Software einem Kenner der FORTRAN Sprache nichts im Wege. Die Grundlage eines CAMAC Aufrufes bestand im ersten Ausbau aus einem bis zu 9 Parametern angewachsenen Unterprogramm-aufruf.

Beispiel:

```
CALL CAMAC (IRWC,ICFU,ILSD,IMOD,ICOM,IFLA,IDAT,IEXP,NADC)
```

Erläuterung der Parameter im Kapitel 5.

Diese sehr umständliche Art, CAMAC anzusprechen wurde wesentlich vereinfacht.

An dieser Stelle zwei Beispiele fuer den vereinfachten Aufruf :

a)     CALL DACN (INUM, IDAT)

Erläuterung: Es soll der DAC mit der Nummer INUM mit dem Wert IDAT beschrieben werden.

b)     CALL ADCN (INUM, IEXP)

Erläuterung: Es soll der ADC mit der Nummer INUM ausgelesen werden. Anweisung fuer den Ort der Datenablage ist die Experimentbezeichnung durch IEXP.

## 10 Programmentwicklung mit Hilfe Indirekter Kommando Files

Aus Gruenden des Bedienungskomforts, Speicherplatzoptimierung und zur Vermeidung von Fehlbedienungen der einzelnen Programm-entwicklungs- Tasks, ist bei diesem System eine indirekte Befehlsdurchfuehrung, die vom Betriebssystem angeboten wird in grossem Masse eingesetzt worden. Sowohl zur Neuerstellung wie auch hauptsaechlich zur Modifizierung existierender Programme, bedient man sich eines Standard- Indirect- Command Satzes.

Allgemeine Erlaeuterungen zu unten aufgelisteten Beispielen

### 10.1 Erste Ebene \*\*\*\*\*

- a) Laufendes Programm beenden
- b) Alle alten Programmversionen loeschen wie :
  - .OBJ
  - .LST
  - .TSK
- c) ASSEMBLER oder FORTRAN Uebersetzungslauf starten
- d) Wahlweiser Ausdruck des LST Files auf dem Printer
- e) Aufruf eines geschachtelten Indirekt Files fuer TKB
- f) Entnahme der alten TASK aus der System Tasktabelle
- g) Einbau der neuen Task in die System Tasktabelle
- h) Start des neuen bzw. modifizierten alten Programms
- i) Ersetzen des OBJ- Files in der U.P. Bibliothek ABEA

### 10.2 Zweite Ebene \*\*\*\*\*

- a) Angaben an den TKB. Filespezifizierungen
- b) Zusatzangaben an den TKB

COMMON  
Globale Spezifikationen  
usw.

10.3 Beispiel 1  
\*\*\*\*\*

FORTRAN Steuerprogramm der TRITIUM Anlage

```
PIP TRIT.LST;*,TRIT.OBJ;*,DL:[1,54]TRIT.TSK;*/DE
FOR TRIT,TRIT/LI:1=TRIT
LST
TKB @TRITT
REM ...TRI
INS DL:[1,54]TRIT/PAR=USER/TASK=...TRI
```

```
DL:[1,54]TRIT/CP=TRIT,ABEA/LB
/
COMMON=EXCOM2:RW
COMMON=EXCOM3:RW
//
```

Beispiel 2  
\*\*\*\*\*

Allgemeines FORTRAN Unterprogramm

```
PIP DANU.OBJ;*,DANU.LST;*/DE
FOR DANU,DANU/LI:1=DANU
LST
LBR ABEA/RP=DANU
```

Beispiel 3  
\*\*\*\*\*

C A M A C TASK

```
PIP CAMA.LST;*,CAMA.OBJ;*,DL:[1,54]CAMA.TSK;*/DE
MAC CAMA,CAMA=MACA.MAC,CAMA
LST
TKB @CAMAT
ABO CAMA
REM CAMA
INS DL:[1,54]CAMA/PRI=220/PAR=USER
RUN CAMA
```

```
DL:[1,54]CAMA/PR=CAMA,SUBLIB/LB
/
COMMON=EXPAGE:RW
//
```

Beispiel 4  
\*\*\*\*\*

ASSEMBLER INTERRUPT SERVICE ROUTINE

```
ABO DAH1
PIP DAH1.OBJ;* ,DAH1.LST;* ,DL:[1,54]DAH1.TSK;* /DE
MAC DAH1,DAH1=DAH1
LST
TKB @DAH1T
REM DAH1
INS DL:[1,54]DAH1/PRI=200/PAR=INT1
RUN DAH1
```

```
DL:[1,54]DAH1/MM/PR:0/-FP,DAH1/-SP/MA=DAH1
DL:[1,54]RSX11M.STB/SS
/
GBLDEF=$VECTR:470
COMMON=EXCOM3:RW
//
```

Beispiel 5  
\*\*\*\*\*

Erstellung einer COMMON TASK

```
MAC EXCOM1,EXCOM1=EXCOM1
LST
TKB @COMB1.CMD
INS DL:[1,1]EXCOM1/PAR=EXCOM1
```

## 11        Datenverwaltung

### 11.1      COMMON Bereiche

Die bei der Systemgenerierung festgelegten COMMON Bereiche werden wie folgt implementiert :

#### 11.1.1    Erstellung

Die Erstellungsprogramme der COMMON Bereiche sind in ASSEMBLER geschrieben.  
Beispiel sei der COMMON Bereich I

Programmlisting der EXCOM1 COMMON Task  
\*\*\*\*\*

```
.PSECT    EXP1,GBL,D,OVR
.BLKW     4095.
.END
```

Erlaeuterung:

Der COMMON soll eine 4K Partition ausfullen.  
Achtung! Laengenangabe : 4095 Worte

#### 11.1.2    Installierung

Die uebersetzte und gebundene COMMON TASK muss in die entsprechende COMMON Partition eingelagert werden.

Kommando : INS DL:[1,1]EXCOM1/PAR=EXCOM1

#### 11.1.3    Initialisierung eines COMMON Bereiches

Jedem COMMON Bereich ist ein Header (Kopf) zugeordnet, der die notwendigen Informationen ueber den Zustand und die Daten des COMMON enthaelt.  
Der Header ist 16 Worte lang.

Informationen :	Tag	\	
	Monat	>	Erstellung
	Jahr	/	
	IDEL		Anzahl der Worte pro Messpunkt

INDE:     Datenverwaltungspointer im COMMON

Zusaetzlich koennen weitere Informationen abgelegt sein.

Art des COMMON Puffers : Normaler oder Ringpuffer  
usw.

Jedem COMMON ist ein Initialisierungsprogramm zugeordnet, das zur Anlaufphase der Experimente gestartet werden muss.

Programme : IC01, IC02, IC03, IC04

#### 11.1.4     Datenverwaltung im COMMON

Entsprechend den Angaben IDEL und INDE aus dem COMMON Kopf wird die aktuelle Adresse fuer die zu speichernden Messdaten ermittelt. Jedem COMMON ist ein Verwaltungsprogramm zugeordnet. Ausserdem koennen die Messwerterfassungsprogramme ihrerseits ueber die Kopfinformationen die Daten selbst verwalten.

Programme : VEC1, VEC2, VEC3, VEC4

#### 11.1.5     Protokollierung der COMMON Daten

Um sich in Kuerze ueber die Daten eines COMMON Bereichs zu informieren, wurden COPY- Tasks implementiert. Sie drucken den aktuellen Kopf des COMMON und die bislang abgespeicherten Messwerte.

Programme : COP1, COP2, COP3, COP4

Da es sich bei den Messdaten oft um Rohdaten handelt, muss zunaechst eine Normierung bzw. Konvertierung durchgefuehrt werden.

Zuweisung der COMMON Bereiche  
\*\*\*\*\*

COMMON I     : Messdaten der ABEA Apparatur

COMMON II    : Messdaten der TRITIUM Apparatur

COMMON III : Konstante und Parameter fuer beide Experimente. Zusaetzlich Zustaende der gesteuerten und adaptierten Hardware ADC's, DAC's, Ventilsteuerung, FLAG-Verwaltung fuer die CAMAC-Synchronisierung.

COMMON IV    : Datenpuffer fuer Auswertungssoftware

## 11.2 DIRECT ACCESS FILES

11.2.1 Zweck der D.A.F.'s ist die Speicherung der Messdaten zu bestimmten Zeitabschnitten auf einem externen Datentraeger fuer spaetere Auswertung, aber auch um genuegend Platz fuer umfangreiche Messreihen zu haben. (COMMON nur 4K Worte)  
Zusaetlich bietet der D.A.F. die Sicherheit, alle bis zu einem eventuell auftretendem Rechnerausfall angefallenen Daten abgespeichert zu haben.

### 11.2.2 Erstellung eines DIRECT ACCESS FILE

Beispiel des CRAD Programm :

```
C*****
C
C      CRAD EROEFFNET DEN FILE > DL1:[3,1]ABEA.DAT
C      BLOCKLAENGE      : 64 WORTE
C      ANZAHL BLOCKS    : 1024
C
C      CALL ASSIGN (1,'DL1:[3,1]ABEA.DAT;1')
C
C      DEFINE FILE 1 (1024,64,U,IREC)
C
C      IREC = 1
C
C      DO 20 K=1,1024
C
20    WRITE (1'IREC) K
C
C      CALL EXIT
C
C      END
```

Der D.A.F ABEA.DAT soll auf der DL1 unter dem UIC [3,1] gefuehrt werden. Er soll maximal 1024 Records enthalten von denen jeder wiederum 64 Worte speichern kann.

### 11.2.3 Initialisierung des D.A.F

Beispiel des Initialisierungsprogramms fuer den ABEA.DAT FILE :

```
C*****
C
C      INTD INITIALISIERT DATENFILE TRIT.DAT
C      RECORD 1 > WORT 1 : NAECHSTER FREIE RECORD
C      STARTWERT : 2
C      RECORD 1 > WORT 2 : NUMMER DER NAECHSTEN MESSUNG
C      STARTWERT : 1
C      RECORD 1 > WORT 3, 4 VORLAEUFIG FREI
C
C      RECORD 1 > WORT 5 : NUMMER DER MESSUNG
C                  > WORT 6 : ANZAHL DER RECORDS
C                  > WORT 7 - 24 MESS- PARAMETER
C
C      RECORD 1 > WORT 25 : NUMMER DER MESSUNG
C                  WORT 26 : ANZAHL DER RECORDS
C                  WORT 27 - 44 MESS- PARAMETER
C
C      RECORD 1 > WORT 45 : NUMMER DER MESSUNG
C                  WORT 46 : ANZAHL DER RECORDS
C                  WORT 47 - 64 MESS- PARAMETER
C
C      RECORD 2 - 1024    > M E S S W E R T E
C
C      CALL ASSIGN (1,'DL1:[3,1]TRIT.DAT;1')
C
C      DEFINE FILE 1 (1024,64,U,IREC)
C
C      READ (1'1) IDUM
C
C      WRITE (1'1) 2, 1
C
C      CALL EXIT
C
C      END
```

In diesem D.A.F koennen maximal Messdaten aus drei verschiedenen Messreihen abgelegt werden. Der erste Record wird als Header benutzt, der alle Informationen ueber den D.A.F enthaelt.

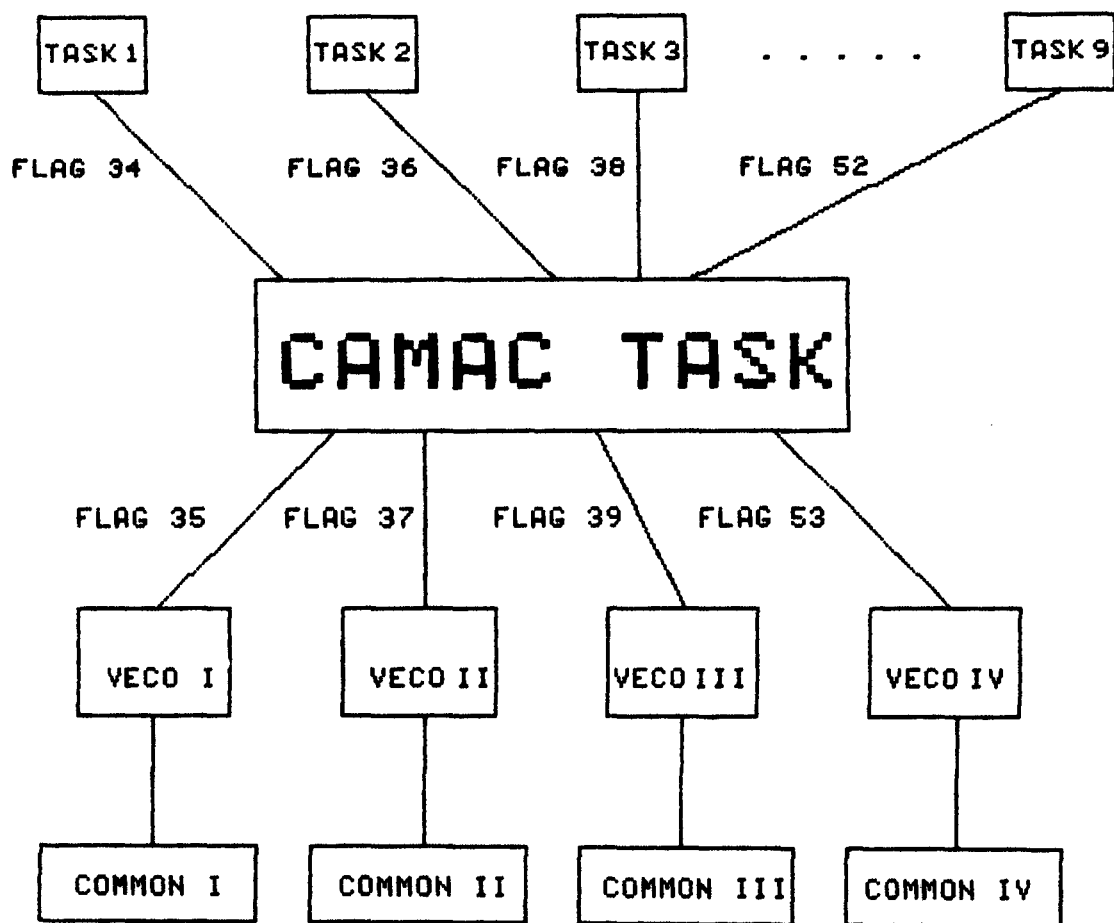
### 11.2.4 Datenverwaltung im D.A.F

Bei der Verwaltung der Messdaten werden die Informationen aus dem Header (1.Record im File) als Parameter verwendet. Gleichzeitig muessen die Header-Informationen aktualisiert werden.



## 12 Software Synchronisierung ueber System Flags

Da die gesamte CAMAC Software (ausser Interruptbehandlung) die sowohl steuernde wie Daten sammelnde Funktionen hat, ueber ein zentrales CAMAC Programm ablaeuft, musste eine Loesung zur Synchronisierung aller CAMAC benutzenden Programme erreicht werden. Dies konnte ueber den vom RSX 11M V3.1 angebotenen FLAG Mechanismus erreicht werden.



Das Diagramm soll diese Synchronisierung andeuten. Die Task, die den CAMAC Handler anspricht, laesst sich ueber eine System FLAG vom Handler die Ausfuehrung der CAMAC Funktion quittieren. Erst dann laeuft sie in ihrem Programmablauf weiter. Die Auswahl der Synchronisierungs-FLAG's geschieht ueber ein Unterprogramm, das aus einer mitverwalteten FLAG- Tabelle FLAG's belegen bzw. wieder freigeben kann. Dies verhindert die gleichzeitige Benutzung der selben System-FLAG von mehreren Programmen. Es handelt sich um die Unterprogramme ALFL und FRFL (siehe Beschreibung der U.P. Bibliothek Kap. 16)

### 13 Abbild des Experiments im COMMON III

Wie schon in 10.1.5 angedeutet, werden im COMMON - Bereich III saemtliche Experiment spezifischen Parameter und Konstante abgelegt. Zusaetzlich werden alle Prozessdaten zum oder vom Prozess als Zwischenpufferung in diesem COMMON verwaltet. Dieser entspricht dem exakten Abbild des rechnergesteuerten Zustandes der beiden zur Zeit angeschlossenen Experimente. Dass zu bestimmten Zeiten dieser COMMON - Bereich auf einen externen Datentraeger kopiert wird, traegt dazu bei, einem eventuellen Rechnerausfall eine abschliessende Analyse des zuletzt festgehaltenen Experimentzustandes folgen zu lassen.

Protokoll des aktuellen COMMON Zustandes:

```
*****
KONSTANTE UND PARAMETER   A B E A
*****
```

ZUSTAENDE DER VENTILE :

```
*****
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
A A A A A A A Z A A Z A A A A A A Z Z Z Z Z Z Z
*****
```

```
AKTUELLE MESSSTELLE   :    24
D A N A - WERT       :    0.000
```

```
ADC NR. : 1  INHALT : 319
ADC NR. : 2  INHALT : 424
ADC NR. : 3  INHALT : 1294
ADC NR. : 4  INHALT : 0
ADC NR. : 5  INHALT : 0
ADC NR. : 6  INHALT : 0
ADC NR. : 7  INHALT : 0
ADC NR. : 8  INHALT : 0
ADC NR. : 9  INHALT : 0
ADC NR. : 10 INHALT : 0
DAC NR. : 1  INHALT : 0
DAC NR. : 2  INHALT : 0
DAC NR. : 3  INHALT : 0
DAC NR. : 4  INHALT : 0
DAC NR. : 5  INHALT : 3000
```

\*\*\*\*\*  
KONSTANTE UND PARAMETER     T R I T  
\*\*\*\*\*

D A N A   -   W E R T            :       16.600

ADC NR. :	1	INHALT :	0
ADC NR. :	2	INHALT :	0
ADC NR. :	3	INHALT :	0
ADC NR. :	4	INHALT :	0
ADC NR. :	5	INHALT :	0
ADC NR. :	6	INHALT :	0
DAC NR. :	1	INHALT :	0
DAC NR. :	2	INHALT :	0
DAC NR. :	3	INHALT :	0

#### 14 Effektivtemperaturbestimmung

Als Beispiel fuer eine vorbereitende Messung soll hier einmal die Bestimmung der Effektivtemperatur des Probenvolumens besprochen werden. Hierbei geht es um folgendes Problem:

Fuer ein ideales Gas, als das der Wasserstoff naeherungsweise angesehen werden kann, gilt folgende Beziehung:

$$p * V = n * R * T$$

p - Druck des Gases

V - Volumen der Gasmenge

n - Gasmenge (Zahl der Mole)

T - Temperatur des Gases in Kelvin

R - allgemeine Gaskonstante

Das bedeutet, dass fuer eine bestimmte Menge an Wasserstoff (n1) in einem festen Volumen (V1) der Druck in folgender Weise von der Temperatur abhaengt:

$$p = (n1 * R / V1) * T \quad \text{oder}$$

$$p = K1 * T$$

Dies ergibt also eine lineare Beziehung des Drucks zur Temperatur.

Probleme ergeben sich, wenn es in dem abgeschlossenen Volumen, in dem der Druck gemessen wird, an verschiedenen Stellen unterschiedliche Temperaturen gibt, d.h. ein Temperaturgradient vorliegt. Nun gilt die lineare Beziehung der allgemeinen Gasgleichung zwischen dem Druck und der an einer (!) Stelle gemessenen Temperatur nicht mehr.

Da es im betreffenden Volumen verschiedene Temperaturen gibt, muss eine effektive Temperatur bestimmt werden, fuer die die Gasgleichung wieder gilt. Diese Effektivtemperatur ist fuer das gewaehlte Volumen und die apparative Anordnung also z.B. Ort des Messthermoelements, Aufbau und Groesse des Volumens eine charakteristische Groesse. Es gilt, ihre Abhaengigkeit von der gemessenen Temperatur zu bestimmen.

Dazu wird zunaechst eine bestimmte Menge an Gas bei Raumtemperatur in das Probenvolumen eingelassen. Bei dieser Temperatur gibt es keinen Temperaturgradienten innerhalb des Probenvolumens, und es muss gelten:

$$T_{\text{eff}} = T_r = T_m$$

$T_{\text{eff}}$  - Effektivtemperatur;  $T_r$  - Raumtemperatur

$T_m$  - gemessene Temperatur

Der Druck bei dieser Temperatur ( $p_r$ ) wird festgehalten und die Temperatur in Schritten erhöht. Nun steigt mit der gemessenen Temperatur auch der Druck und es ergibt sich folgende Beziehung:

$$T_{eff} (T_m) = ((p_m - p_r) / p_r) * T_r + T_r$$

$T_{eff}$  - Effektivtemperatur  
 $T_m$  - gemessene Temperatur  
 $p_m$  - gemessener Druck  
 $p_r$  - Druck bei Raumtemperatur  
 $T_r$  - Raumtemperatur

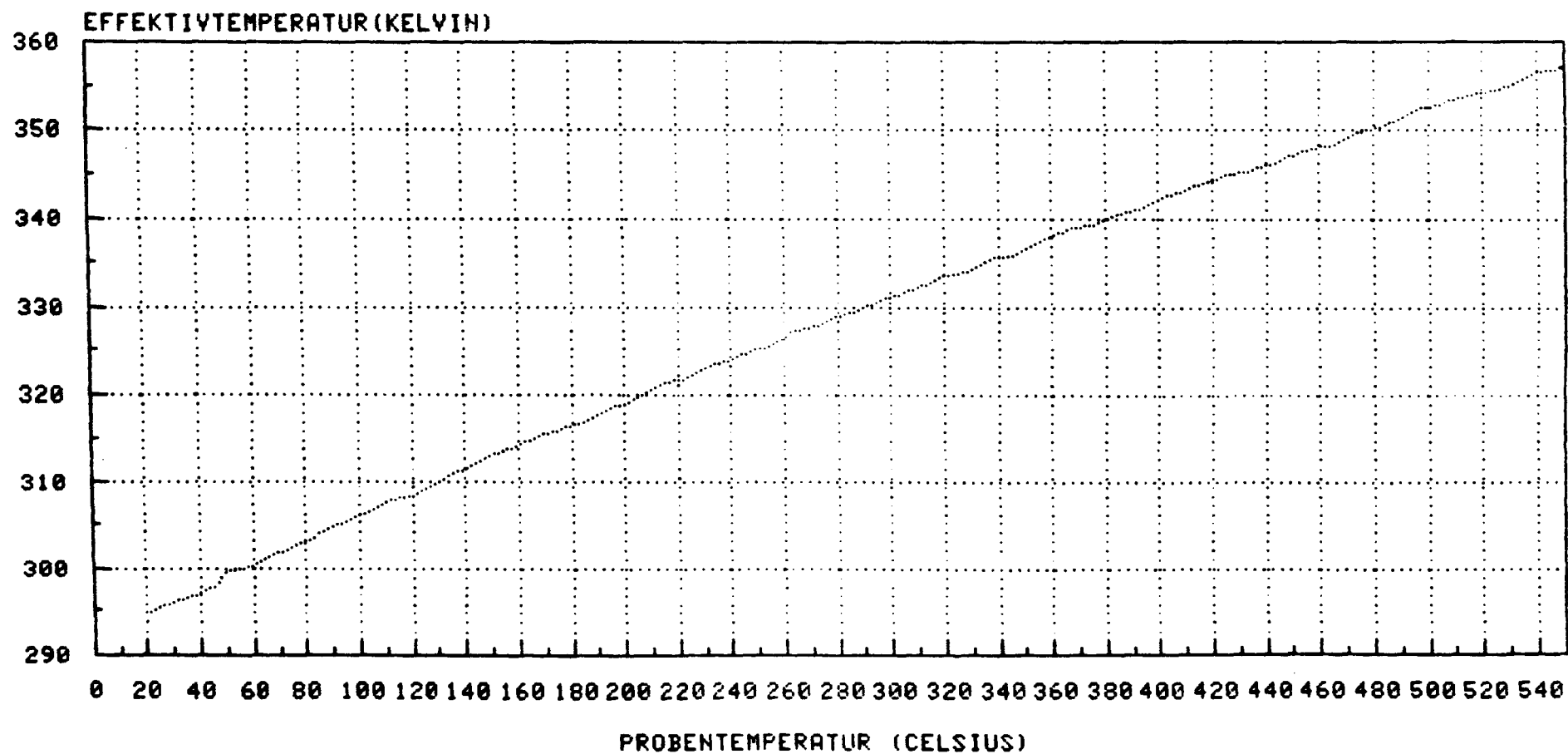
So ergibt sich eine Abhängigkeit der Effektivtemperatur von der gemessenen Temperatur.

Die Kenntnis der Effektivtemperatur ist von grosser Wichtigkeit, um aus den bei einer Beladung gemessenen Druckabfällen die Wasserstoffkonzentration in der Probe zu bestimmen.

Die hier beschriebene Messung wurde mit Hilfe des Rechners automatisch durchgeführt und ausgewertet. Das Ergebnis ist in Abb. 6 zu sehen.

# EFFEKTIVTEMPERATUR: PROBENVOLUMEN

MECKING 02-MAY-80



## 15. Tritium-Messprogramm

Als Beispiel fuer ein Mess- und Steuerprogramm soll hier einmal das Programm zur Steuerung und Datenerfassung der Tritium-Anlage besprochen werden.

```
C*****
C
C      PROGRAMNAME          :          TRIT.FTN
C
C      BEARBEITER           :          H. HEER, P.MECKING
C
C      ERSTELLUNG           :          08-NOV-79
C
C      LETZTE AENDERUNG     :          20-MAR-80
C
C*****
C
C      DATENERFASSUNG UND REGELUNG DER TRITIUMANLAGE
C      DATEN ARCHIVIERUNG AUF DL1:[3,1]TRIT.DAT;1
C
C      *****
C      * C O M M O N - B E R E I C H E
C      *****
C      *
C      COMMON /EXP2/ KOPF(16), IFEL (4032)
C      COMMON /EXP3/ KOP1(16), KOPA1 (64)
C      1          , KOP2 (16), KOPA2 (64)
C      1          , IFAF (10)
C
C      *
C      *****
C
C      INTEGER ANF, ZEIT, IDAT (2), IBCD (2), INTE (2), ITIM (4)
C      INTEGER INPUF (7), IFIL (64), INRE (64), TRIT
C      REAL DTES (2), ICO2, TEMTE(2)
C      LOGICAL *1 JA, ICM, IAB
C
C      EQUIVALENCE (IDAT(1),IVAR2), (IDAT(2),IVAR1)
C      EQUIVALENCE (KOPF(4),IDEL), (KOPF(5),INDE)
C      EQUIVALENCE (INRE(1),IREC), (INRE(2),ICM)
C      1, (INRE(3),ANF), (INRE(4),KSTEP), (INRE(5),ITMAX)
C      2, (INRE(6),IBAR), (INRE(7),ZEIT), (INRE(8),IWEIL)
C      3, (INRE(9),KOTEN)
C
C      DATA VEC3 /6RVEC3 /, TRIT /2/
C      DATA ICO2 /6R...IC2/
C      DATA JA /"112/
```

```
C *****
C AKTIVIERE DATENVERWALTUNGSPROGRAMM
C *****
C
C CALL REQUES (VEC3)
C
C CALL ASSIGN (1,'DL1:[3,1]TRIT.DAT;1')
C DEFINE FILE 1 (1024,64,U,IROC)
C
C CALL ASNLUN (6,'TT',0)
C
C CALL CLEAR
```

Der Anfang des Programms beinhaltet die Datendefinitionen und -ueberlagerungen, sowie das Starten des Datenverwaltungsprogramms fuer den Common III und das Assignment des Datenfiles, auf dem die Messdaten abgelegt werden sollen.

```
C
C *****
C P R O G R A M M S T E U E R U N G S - D I A L O G
C *****
C
C CALL GONG
C WRITE (6,75)
75 FORMAT ('$WELCHES TERMINAL ? ( TT0=0 , TT1=1 ) : ')
C
C READ (6,76,END=200) NR
76 FORMAT (I1)
C
C CALL GONG
C WRITE (6,77)
77 FORMAT ('$MESSWERTE STAENDIG DRUCKEN ? (JA = J) : ')
C
C READ (6,78,END=200) IAB
78 FORMAT(A1)
C
C CALL GONG
C WRITE (6,83)
83 FORMAT ('$TRIT.-DATEN INITIALISIEREN ? (JA = J) : ')
C
C READ (6,84,END=200) ICM
84 FORMAT (A1)
C
C CALL GONG
115 WRITE (6,400)
400 FORMAT ('$GIB STARTTEMPERATUR (200 - 560 CEL.) : ')
C
C READ (6,500,END=200) ANF
500 FORMAT(I4)
C
C IF (ANF .GT. 560) GO TO 115
C
C CALL GONG
126 WRITE (6,600)
600 FORMAT ('$GIB TEMPERATURSCHRITT (0 - 100 CEL.) : ')
```



```

700  READ (6,700,END=200) ITEMS
      FORMAT (I3)
C
      IF (ITEMS .GT. 100) GOTO 126
C
      KSTEP = ITEMS
C
      CALL GONG
      WRITE (6,90)
90    FORMAT ('$GIB MAXIMALTEMPERATUR ( CELSIUS ) : ')
C
      READ (6,91,END=200) ITMAX
91    FORMAT (I4)
C
      CALL GONG
      WRITE (6,92)
92    FORMAT ('$WELCHES BAROCEL ? (1000 ODER 10 TORR) : ')
C
      READ (6,93,END=200) IBAR
93    FORMAT (I4)
C
      CALL GONG
      WRITE (6,800)
800   FORMAT ('$GIB MESSZEITINTERVALL ( 1 - 999 SEK ) : ')
C
      READ (6,900,END=200) ZEIT
900   FORMAT (I3)
C
      CALL GONG
      WRITE (6,94)
94    FORMAT ('$WARTEZEIT NACH DEM HEIZEN ( 0-99 MIN) : ')
C
      READ (6,95,END=200) IWEIL
95    FORMAT (I2)
C
      CALL GONG
      WRITE (6,96)
96    FORMAT ('$GIB ENDTEMPERATUR (ABKUEHLUNG) : ')
C
      READ (6,97) ITEMEN
97    FORMAT (1 I4)
```

Im Programmsteuerungsdialog mit dem Benutzer werden die Messparameter abgefragt und eingelesen.

```
C      *****
C      KONVERTIERUNG DER EINGABEN
C      *****
C
C      ANF = IFIX ((ANF + 93)/0.413)
C
C      ISTEP = IFIX (ITEMS * 2.5)
C
C      KOTEN = IFIX ((ITEMEN + 93)/0.413)
```

Die Eingaben des Benutzers bezueglich den Temperaturen  
werden zur Programmierung des DAC konvertiert.  
Die Formel ist geraetespezifisch fuer die Probenheizung.

```
C      *****
C      BELEGE SYSTEM - FLAG
C      *****
C
C      CALL ALFL (IFLA)
C
C      IF (IFLA .EQ. -1) GO TO 6666
```

Es wird eine Systemflag zum Dialog mit dem Handler und  
dem Datenverwaltungsprogramm belegt.

```
C
C      *****
C      MESSPARAMETER AUF DIR. ACC. FILE
C      *****
C
C      READ (1'1) IREC
C
C      IF (IREC .GT. 1024) GO TO 6000
C
C      WRITE (1'1) (INRE(K), K=1,64)
C
```

Die Messparameter werden auf den Direct-Access File auf der  
Platte geschrieben.

```
*****
INITIALISIERE TEMPERATUR KOEFFIZIENT
*****

KOTE = ANF
```

```
C *****
C D A C 6  INITIALISIEREN
C *****
C
C   IDAT (2) = KOTE
C   NUDA = 9
C   KOPA2 (NUDA + 30) = IDAT (2)
C
C   CALL DACN (NUDA, IDAT)
C
```

Der DAC wird mit dem Anfangswert beschrieben, und so die Anfangstemperatur eingestellt.

```
C
C *****
C TEMP.  AUS EXCOM3  AUSLESEN
C *****
C
1000 IF ((KOTE.LT.KOTEN) .AND. (ISTEP .LT. 0)) GOTO 7000
C
C *****
C 6 * PARAMETER MESSEN
C *****
C
C   DO 10 I1= 1, 6
C
C   IBCD (1) = KOPA2 (27)
C   IBCD (2) = KOPA2 (28)
C
C   CALL BCIN (IBCD, INTE)
C
C   RV1 = INTE(1)
C   RV2 = INTE(2)
C   ERG = RV1 + RV2/10000.
C   ERG = ERG*1000.
C
C   CALL TIME (ITIM)
C
C   INX = ((INDE-1)*IDEL) + 1
C
C   IFEL (INX) = ITIM (1)
C   INX = INX+1
C   IFEL (INX) = ITIM (2)
C   INX=INX+1
C   IFEL (INX) = ITIM (3)
C   INX = INX+1
C   IFEL (INX) = ITIM (4)
C   INX = INX+1
C   IFEL (INX) = KOPA2 (27)
C   INX = INX+1
```

```

IFEL (INX) = KOPA2 (28)
INX = INX+1
IFEL (INX) = KOTE
INX = INX+1
C
C *****
C LIES JOERGER ADC DRUCK
C *****
C
NUAD = 41
C
CALL ADCN (NUAD,TRIT)
IFEL (INX) = KOPA2 (NUAD-12)
C
DRUCK = (IFEL (INX)/4095.) * FLOAT (IBAR)
REZT = 1./(ERG + 273.15)
C
IF (IAB .NE. JA) GOTO 81
C
WRITE (6,100) (ITIM(K),K=1,4), ERG, DRUCK,REZT
100 FORMAT (/X,4A2,5X,F9.1,3X,5X,F7.1,5X,E10.4)
C
C *****
C NAECHSTER MESSPUNKT TRITIUM ?
C *****
C
81 INDE = INDE + 1
C
C *****
C MESSDATEN AUF DIRECT ACCESS FILE
C *****
C
READ (1'1) (INRE(K), K=1,64)
C
IANF = ((INDE-9)*IDEL) + 1
IEND = IANF + 63
C
WRITE (1'IREC) (IFEL(K), K=IANF, IEND)
C
C *****
C DATEN FILE VOLL
C *****
C
IREC = IREC + 1
C
WRITE (1'1) (INRE(K), K=1,64)
```

```
C      *****
C      REAL KONVERTIERUNG DER EINGABEN
C      *****
C
C      RBAR = FLOAT(IBAR) - FLOAT(IBAR)/500
C
C      TMAX = FLOAT(ITMAX)
C
C      *****
C      ABFRAGE WERTUEBERSCHREITUNG. FALLS
C      T > 300 C - REGELMECHANISMUS
C      *****
C
C      IF (KOTE .GE. 1700) GOTO 3000
C      IF (ERG .GE. TMAX) GOTO 3000
C      IF (DRUCK .GE. RBAR) GOTO 3000
C      IF (ERG .GE. 300.) GOTO 5000
C
C      CALL MARK (1,ZEIT,2)
C      CALL WAITFR (1)
10    CONTINUE
```

Im vorstehenden Teil des Programms wird folgendes durchgefuehrt:

1) Es wird sechsmal im Abstand einer vorgegebenen Zeitspanne Druck und Temperatur gemessen, ausgeschrieben und im Common abgelegt. Ebenfalls werden die Zeit und ein wichtiger Messparameter (Kote) im Common gespeichert.

Nach jeweils acht Messzyklen werden die Messwerte aus dem Common auf den DA-File geschrieben. Der Common wird als Ringpuffer verwendet, bei Ueberlauf wird er von vorn ueberschrieben.

2) Es sind Sicherheitsabfragen eingebaut, die sicherstellen, dass der Druck bzw. die Temperatur einen vorgegebenen Grenzwert nicht uebersteigen koennen. Bei Annaeherung an diesen Grenzwert wird die Temperatur erniedrigt.

3) Steigt die Temperatur ueber 300 grd Cels., so soll ein neuer Entscheidungsmechanismus in Kraft treten, der feststellt, ob die Temperatur erhoeht werden kann.

```
      GOTO 4000
C
3000 IF (ISTEP .LT. 0) GOTO 8000
C
      WRITE (6,1100)
1100 FORMAT (/X,'      TEMP.-ABSENKUNG')
C
      ISTEP = -ISTEP
C
      GOTO 4000
```

Der vorstehende Teil wird erreicht, wenn eine Wertueberschreitung vorliegt. Die Groesse ISTEP, die den Temperaturschritt angibt, wird negativ, was bedeutet, dass die Temperatur nun abgesenkt wird.

```
C      *****
C      TEMPERATUR SENKEN AUF 0
C      *****
C
2000  KOTE = 0
      IDAT (2) = KOTE
      KOPA2 (NUDA + 30) = IDAT (2)
C
      CALL DACN (NUDA, IDAT)
C
      WRITE (6,550)
550   FORMAT (/X,' ***** M E S S U N G       B E E N D E T *****')
      CALL GONG
C
      GOTO 203
```

Dieser Programmteil schaltet nach Beendigung der Messung den Regler aus.

```
C      *****
C      TEMPERATURREGELUNG : WARTEN BIS DELTA T < 0.2 %
C      *****
C
565   DO 17 I=1,2
C
      IBCD(1) = KOPA2(27)
      IBCD(2) = KOPA2(28)
C
      CALL BCIN (IBCD,INTE)
C
      RV1 = INTE (1)
      RV2 = INTE (2)
C
      ERG = RV1 + RV2 /10000.
      ERG = ERG * 1000.
C
      TEMTE(I) = ERG
C
      CALL ADCN (NUAD, TRIT)
C
      DTES (I) = KOPA2 (NUAD-12)
C
      IF (DTES(I) .GE. 4090.) GOTO 3000
      IF (ERG .GE. TMAX) GOTO 3000
C
      CALL MARK (1,2,3)
      CALL WAITFR (1)
```

```
17    CONTINUE
C
      DTEM = (TEMTE(2) - TEMTE(1)) / TEMTE(1)
C
      IF ( ABS(DTEM) .GE. 0.002) GOTO 565
```

Dieser Programmteil tritt bei Temperaturen ueber 300 grd Cels. in Kraft. Der Rechner liest im Abstand von zwei Minuten die Temperatur aus und bestimmt, ob der Unterschied der beiden Messwerte kleiner als 0.2% ist. Ist dies nicht der Fall, wiederholt er die Operation.

```
553    DO 15 I=1,2
C
      CALL ADCN (NUAD, TRIT)
C
      DTES(I) = KOPA2 (NUAD-12)
C
      IBCD(1) = KOPA2 (27)
      IBCD(2) = KOPA2 (28)
C
      CALL BCIN (IBCD,INTE)
C
      RV1 = INTE (1)
      RV2 = INTE (2)
C
      ERG = RV1 + RV2 /10000.
      ERG = ERG * 1000.
      IF (ERG .GE. TMAX) GOTO 3000
C
      CALL MARK (1,2,3)
      CALL WAITFR (1)
C
15    CONTINUE
C
      IF (DTES(1) .EQ. 0.) DTES(1)=1.
C
      DELTDR = (DTES(2) -DTES(1))/DTES(1)
C
551    IF (ABS(DELTDR) .GT. 0.002) GOTO 553
```

Der gleiche Entscheidungsprozess wird hier fuer den Druck vorgenommen. Ist die Druckaenderung in zwei Minuten kleiner als 0.2%, so ist bei der Begasung ein Gleichgewichtszustand erreicht und die Temperatur kann fuer den naechsten Messpunkt erhoehrt bzw. erniedrigt werden.

```
4000 KOTE = KOTE + ISTEP
      IDAT (2) = KOTE
      KOPA2 (NUDA + 30) = IDAT (2)
C
      CALL DACN (NUDA, IDAT)
C
      GOTO 1000
```

Hier wird die Temperatur erhoeht, bzw. abgesenkt

```
200  WRITE (6,201)
201  FORMAT (/X,' ***** PROGRAMMABBRUCH BEI DER EINGABE *****')
      CALL GONG
C
      GOTO 2000
C
6000  WRITE (6,6010)
6010  FORMAT (/X,' ***** PROGRAMMABBRUCH F I L E VOLL *****')
      CALL GONG
C
      GO TO 2000
C
C *****
C SCHREIBE DIE LETZTEN MESSWERTE
C *****
C
7000  IANF = IEND + 1
      IEND = IANF + 63
C
      READ (1'1) (INRE(K), K=1,64)
C
      WRITE (1'REC) (IFEL(K), K=IANF,IEND)
C
      IREC = IREC + 1
C
      WRITE (1'1) (INRE(K), K=1,64)
C
      GO TO 2000
C
203  CALL FRFL (IFLA)
C
8888 CALL EXIT
C
      END
```

In diesem Programmteil werden nach Beendigung der Messung die letzten noch verbliebenen Messwerte auf den DA-File geschrieben und die System-Flag wieder freigegeben.



## 16 Test - Software

Weil es sich in der Praxis immer bewahrt hat, modulare Testsoftware zu konzipieren, wird auch dieses Programmpaket von fundamentalen Testprogrammen unterstuetzt. Hierbei kann schrittweise einem schwer zu lokalisierenden Fehler nachgegangen werden.

Zunaechst lassen sich die benutzten CAMAC Module als einzelne Geraete testen. In diesem Sinn laesst sich auch die eigens fuer dieses Experiment entwickelte Ventilsteuerung ueberpruefen. Die Messstellenumschaltung laesst sich auch getrennt steuern. Des weiteren ist ein simulierter Testbetrieb fuer beide Experimente implementiert, bei denen in ueberschaubaren Zeitraeumen saemtliche Funktionen der Experiment-Apparatur ueberprueft werden koennen. Getrennt davon existieren Testprogramme, die im Entwicklungsstadium der Anwendersoftware entstanden sind und festgehalten wurden, weil sie in Zukunft bei Stoerfaellen wieder ihren Einsatz finden.

### 16.1 ADC Testprogramm

Dieses Programm dient zum Test der JOERGER ADC's. Es initialisiert das Kontrollwort des ADC, d.h. ob 1, 2, 3, 4 oder alle 16 ADC's abgefragt werden sollen.

Bei festvorgegebener Spannung (0 - 10Volt) an einem ausgewaehlten Eingang wird der Wert ins Digitale konvertiert ausgelesen. Da diese modulare Testprogramme der ADC Hardware als privilegierte Tasks in ASSEMBLER geschrieben sind, haben sie direkten Zugriff auf CAMAC und schliessen so Fehler in der Umgebungssoftware aus. (CAMAC Driver oder Synchronisierung etc.)

### 16.2 DAC Testprogramm

Das Programm testet den KINETIC DAC. (8 DAC's im Modul) Auch dieses Programm ist als privilegierte ASSEMBLER Task autonom (d.h. Ausschluss des CAMAC Handler und Synchronis.)

Digital werden im Bereich von 0 bis 4095 Zahlen in den gewuenschten DAC eingeschrieben, die an den entsprechenden DAC Ausgaengen jeweils Spannungen von 0 bis 10 Volt erzeugen sollten.

Die oben beschriebenen Programme ADC-Test und DAC-Test lassen sich optimal kombinieren, indem man die Ausgaenge der DAC's mit den Eingaengen der ADC's koppelt. So hat man einen geschlossenen Kreis und kann den gesamten Messbereich beliebig genau durchfahren. (max. 4096 Schritte)

### 16.3 Test der TELEFUNKEN Parallel-Output-Register

Das Programm ist als privilegierte ASSEMBLER- Task geschrieben. Es kann gezielt jede gewuenschte Bit-Konfiguration ins POR schreiben, oder jedes einzelne Bit ein- oder ausschalten. An der Ausgangsseite laesst sich ein TELEFUNKEN Input-Register anschliessen. Somit waere der Testkreis geschlossen. Treten Fehler auf, loest man die Verbindung und testet gezielt Input bzw. Output Seite.

### 16.4 Messstellenumschaltung - Testprogramm

Wesentlich fuer den spaeteren Prozessablauf ist die Funktions-tuechtigkeit des Messstellenumschalters. Das Programm arbeitet im Dialog mit dem Anwender. Von der aktuell angefahrenen Messstelle soll auf eine andere Messstelle umgeschaltet werden. Es koennen maximal 24 Messstellen angeschlossen werden. Zur Zeit sind 4 Messstellen in folgender Reihenfolge angeschlossen, um effektiv umschalten zu koennen:

1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4 ..... , 1, 2, 3, 4

### 16.5 Ventilsteuerung Testprogramm

Prozessteuernd greift der Rechner hauptsaechlich auf die Ventilsteuerung der ABEA - Anlage ein. Somit sollte mit Sicherheit geprueft werden, dass sich alle Ventile gezielt oeffnen und schliessen lassen. Dieses ermoeoglicht das in FORTRAN geschriebene Testprogramm. Es arbeitet im Dialogverkehr.

Bedienungsbeispiel:

>VENT

GIB VENTILNUMMER (0 - 24) : 23  
VENTIL OEFFNEN (JA = J) : J

GIB VENTILNUMMER (0 - 24) : 21  
VENTIL OEFFNEN (JA = J) : N

Mit einem aehnlichen Programm wurde in einem Langzeittest ein Ventil ueber drei Tage ca. 30000 mal betaetigt.

## 16.6 Experimentsimulation

Als Gesamttest bestehen fuer beide Experimente Simulationsprogramme. Diese sollen eine Ueberpruefung fast aller Komponenten in ueberschaubaren Zeitintervallen gewaehrleisten. Es werden alle Funktionen des spaeteren Experimentablaufs durchgefuehrt. Die Datenverwaltung laeuft sowohl in den entsprechenden COMMON Bereichen wie auch in den zugehoerigen Direct Access Files. Somit koennen auch Testprogramme fuer die Auswertung und Darstellung auf die exakten Quelldaten zugreifen, so wie es spaeter im Experimentierbetrieb sein wird. Ein weiterer Test ist bei Ablauf beider Programme auch fuer die Synchronisierung ueber die System FLAGS und natuerlich auch fuer die Flagverwaltung gegeben.

## 16.7 Testprogramm System-Flag-Verwaltung

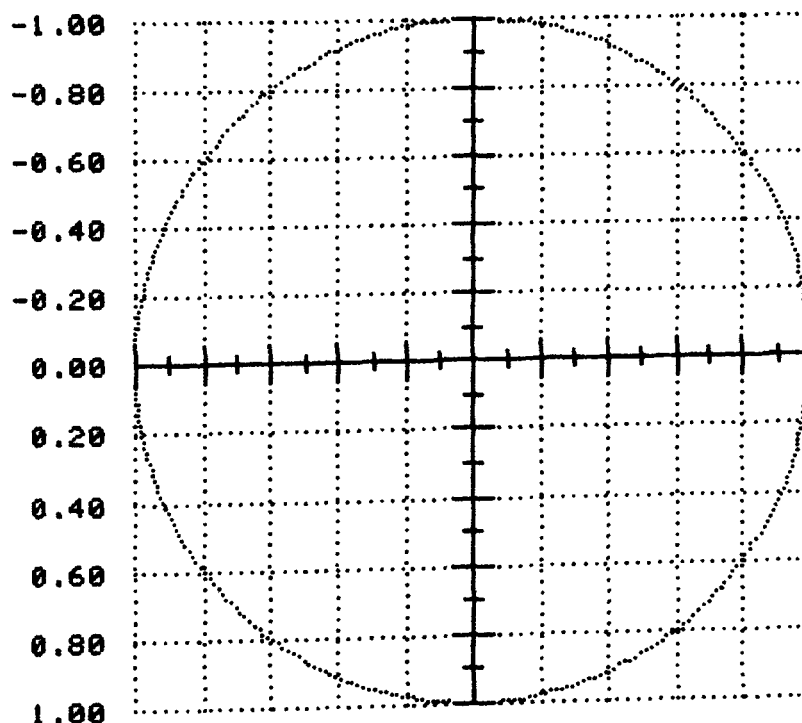
Der aktuelle Status der System Flags kann durch Aufruf der BEFL Task ueberprueft werden.

## 16.8 HP Graphik BASIC Testprogramm

Testprogramme fuer die Kommunikation PDP 11/34 und HP 2647/A

ES wurden in FORTRAN einige mathematische Kurven tabellarisch errechnet, (SINUS, COSIN, KREIS, usw.) deren Daten bei Aufruf vom OFF LINE betriebenen HP Terminal vom Rechner an dieses gesendet werden. Skalierungsdaten ermöglichen die maximale Ausnutzung des HP-Display fuer die geforderte graphische Darstellung.

Beispiel : KREIS - Funktion



## 17 Unterprogrammbibliothek : A B E A . O L B

Kurzbeschreibung der Unterprogramme in alphabetischer Reihenfolge.

Zur einfachen Handhabung und zur besseren Organisation dient letztlich die Zusammenfassung aller im System anfallenden Unterprogramme (U.P.'s in FORTRAN oder ASSEMBLER) in einer U.P. - Bibliothek. Dies bringt besonderen Bedienungskomfort beim Arbeiten mit dem Binderprogramm TKB. Auf den folgenden Seiten sind alle vorhandenen U.P.'e in Kuerze ueber ihren Funktionsablauf beschrieben.

Bemerkung: Einzelne U.P.e stellen modulare Bloecke der gesamten Steuerung und Messwerterfassung dar, und bietet fuer sich selbst oft eine gute Testmoeglichkeit der einzelnen Komponenten.

### 17.1 ADCN

DAS U.P. ADCN liest den geforderten JOERGER ADC in die entsprechende COMMON Zelle aus.

Aufruf : CALL ADCN (NUAD, EXPE)

Parameter : NUAD > Nummer des ADC  
EXPE > Experiment

Intern aufgerufene U.P. : ALFL, CAMAC, FRFL

### 17.2 ALDA

ALDA faehrt alle Messstellen an und uebernimmt die anstehenden Daten.

Aufruf : CALL ALDA

Intern aufgerufene U.P. : DAME

### 17.3 ALFL

Das U.P. ALFL sucht eine freie GLOBAL FLAG. FLAG Verwaltung im COMMON III. Spezifizierte FLAGS :

34, 36, 38, ... ,50, 52

Aufruf : CALL ALFL (IFLA)

Parameter : IFLA > Nummer der freien FLAG

Achtung : -1 >>> zur Zeit keine FLAG frei

#### 17.4 BAPA

BAPA sucht in einem Datenfeld nach dem minimalen und maximalen Wert.

Aufruf : CALL BAPA (LANG, FELD, RMIN, RMAX)

Parameter : LANG > Laenge des Datenfeldes  
FELD > Datenfeld  
RMIN > Minimum - Realwert  
RMAX > Maximum - Realwert

#### 17.5 BCIN

Konvertierung von maximal 6 BCD Daten in Integer Zahlen

Maximale BCD Ziffer : 199999  
Ergebnis 2 Integer : 19  
9999

Aufruf : CALL BCIN (IBCD, INTE)  
Parameter : IBCD > 2 Worte BCD Zahl  
INTE > 2 Worte Integer Zahl

#### 17.6 CALA

Das U.P. CALA errechnet aus den Kalibrierungsdaten die Verhaeltnisse der Volumina bei der ABEA Anlage.

Aufruf : CALL CALA

Intern aufgerufene U.P. : GONG

#### 17.7 CAMAC

Ueber das U.P. CAMAC wird der Zugriff auf die CAMAC- Hardware durchgefuehrt.

Aufruf : CALL CAMAC (P1,P2,P3,P4,P5,P6,P7,P8,P9)

Parameter : P1 > Read, Write oder Cntr  
P2 > CAMAC Function  
P3 > 24 oder 16 Bit Datentransfer  
P4 > Modulnummer im CRATE (Stationsnummer)  
P5 > Anzusprechender COMMON Bereich  
P6 > Synchronisations - FLAG  
P7 > Daten  
P8 > Experimentbezeichnung  
P9 > Nummer eines ADC's

#### 17.8 CLEAR

CLEAR loescht den HP 2647/A Terminal Display und setzt den Cursor in Home- Stellung. (d.h. links oben)

Aufruf : CALL CLEAR

Intern aufgerufene U.P.'e : ESCA

#### 17.9 CONU

Das U>P. CONU normiert den vom ADC gelieferten Wert auf das geforderte UHV-Druck Massystem.

Aufruf : CALL CONU (IDAT, UDRU)

Parameter : IDAT > ADC- Wert (0-4095)  
UDRU > Druck im geforderten Massystem (scaliert)

#### 17.10 CONV

CONV fuehrt eine Normierung eines ADC- Wertes an Hand einer Eich-tabelle auf das geforderte Vordruck-Masssystem durch.

Aufruf : CALL CONV (IDAT, VORD)

Parameter : IDAT > ADC- Wert  
VORD > Vordruck im geforderten Massystem (scaliert)

#### 17.11 DACN

DAS U.P. DACN beschreibt den geforderten KINETIC DAC.

Aufruf : CALL DACN (NUDA, IDAT)

Parameter : IDAT > Wert fuer die Konvertierung

Intern aufgerufene U.P.'e : ALFL, CAMAC, FRFL

## 17.12 DAME

DAME faehrt eine bestimmte Messstelle an, liest den zur Zeit anstehenden Wert, konvertiert diesen und speichert ihn ab.

Aufruf : CALL DAME (IMES)

Parameter : IMES > Nummer der Messstelle  
1 > Probendruck  
2 > Vorvolumen H2 Druck  
3 > Vorvolumen D2 Druck  
4 > Probentemperatur

Intern aufgerufene U.P.'e : MEUM, WAIT, BCIN

## 17.13 DANU

Das U.P. DANU ist identisch mit DAME . Der errechnete Wert wird nicht im COMMON abgespeichert sondern wird dem aufrufenden Programm zur Verarbeitung uebergeben.

## 17.14 ESCA

ESCA enthaelt alle oft benutzten Escape-Sequenzen des HP 2647 Terminal. Siehe auch Kapitel 6 II (Zur Zeit ca. 80 Sequenzen)

Aufruf : CALL ESCA (NESC)

Parameter : NESC > Nummer der Escape-Sequenz (Kapitel 6 II)

## 17.15 FILA

Das U.P. FILA schaltet das FILAMENT an der Messroehre des Ionisationsmanometers ein bzw. aus.  
Es wird am DAC ein Impuls generiert. (ca. 5V - 20msec)

Aufruf : CALL FILA

Intern aufgerufene U.P.'e : DACN, WAIT

## 17.16 FRFL

Das U.P. FRFL gibt eine globale System Flag frei. Flag- Verwaltung im COMMON III. (Gegenprogramm von ALFL)

Aufruf : CALL FRFL (IFLA)

Parameter : IFLA > Flag Nummer (34 - 52)

17.17 GONG

GONG erzeugt auf dem HP 2647/A einen kurzen Alarmton. Es dient zur Aufforderung zu einer vom Rechner erwarteten Eingabe am Terminal oder schliesst entscheidende Meldungen um Aufmerksamkeit zu erwecken.

Aufruf : CALL GONG

17.18 IONI

IONI liest das Ionisationsmanometer aus und schaltet, falls erforderlich das FILAMENT am Ionisationsmanometer ein.

Aufruf : CALL IONI (UDRU)

Parameter : UDRU = UHV - Druck

Intern aufgerufene U.P.'e : WAIT, ADCN, FILA

17.19 KAPI

KAPI pumpt die Vorvolumina ausser den BAROCEL ab.

Aufruf : CALL KAPI

Intern aufgerufene U.P.'e : VEST, WAIT, IONI

17.20 KAPU

KAPU versetzt die ABEA - Aperatur nach der Kalibrierung in den UHV- Zustand.

Aufruf : CALL KAPU

Intern aufgerufene U.P.'e : ADCN, WAIT, FILA, VEUH

17.21 MEPU

MEPU versetzt die ABEA - Aperatur nach dem Druckeinlass in den UHV - Zustand.

Aufruf : CALL MEPU

Intern aufgerufene U.P.'e : ADCN, FILA, VEUH, WAIT



## 17.22 MEUM

Das U.P. MEUM schaltet den BURSTER Messstellenumschalter auf die gewünschte Station. Stationsverwaltung im COMMON III.

Aufruf : CALL MEUM (NUME)

Parameter : NUME > Nummer der Messstelle  
NUME > -1 d.h. Messstelle < 1 oder > 4 verlangt

Intern aufgerufene U.P.'e : ALFL, CAMAC, WAIT, FRFL

## 17.23 TEMP

TEMP misst die Proben temperatur an der ABEA - Anlage. Das Programm misst 2 mal pro Sekunde insgesamt 5 mal, mittelt die Messwerte normiert sie und gibt sie ans aufrufende Programm zurück.

Aufruf : CALL TEMP (RTEM)

Parameter : RTEM > Gemessene Temperatur (Real - Wert)

## 17.24 VEST

Das U.P. VEST öffnet oder schliesst ein Ventil der ABEA - Apparatur. Im COMMON III werden die Ventil-Zustandslisten verwaltet.

Aufruf : CALL VEST (VENU, VEFU)

Parameter : VENU > Ventilnummer  
VEFU > Ventilfunktion ( 0 = zu 1 = auf )

Intern aufgerufene U.P.'e : ALFL, CAMAC, FRFL

## 17.25 VEUH

VEUH steuert die Ventile der ABEA - Apparatur fuer den UHV-Druck Zustand.

Aufruf : CALL VEUH (NUVE)

Parameter : NUVE > Nummer des Ventils

Intern aufgerufene U.P.'e : WAIT, ADCN, CONU, VEST, FILA

## 17.26 VEVO

VEVO steuert die Ventile der ABEA - Anlage fuer den Vordruck-Zustand.

Aufruf : CALL VEVO (NUVE)

Parameter : NUVE > Nummer des Ventils

Intern aufgerufene U.P.'e : WAIT, ADCN, CONV, GONG, VEST

## 17.27 WAIT

DAS WAIT Unterprogramm setzt einen Warteaufruf ans Betriebssystem auf.

Aufruf : CALL WAIT (IANZ, IEIN)

Parameter : IANZ > Anzahl der Einheiten

IEIN > Zeiteinheit

1 > TICS (20m sec)

2 > Sekunden

3 > Minuten

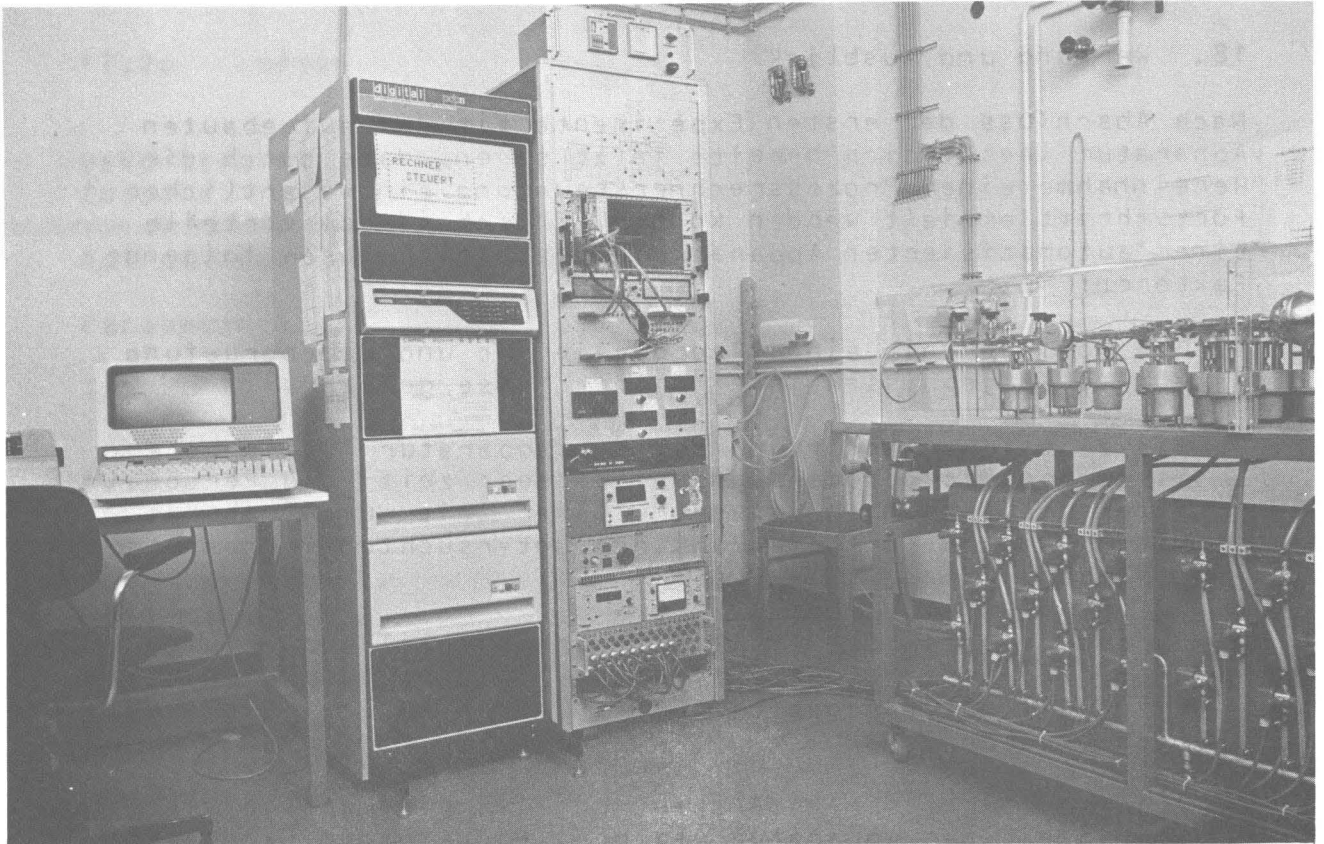
4 > Stunden

## 18. Wertung und Ausblick

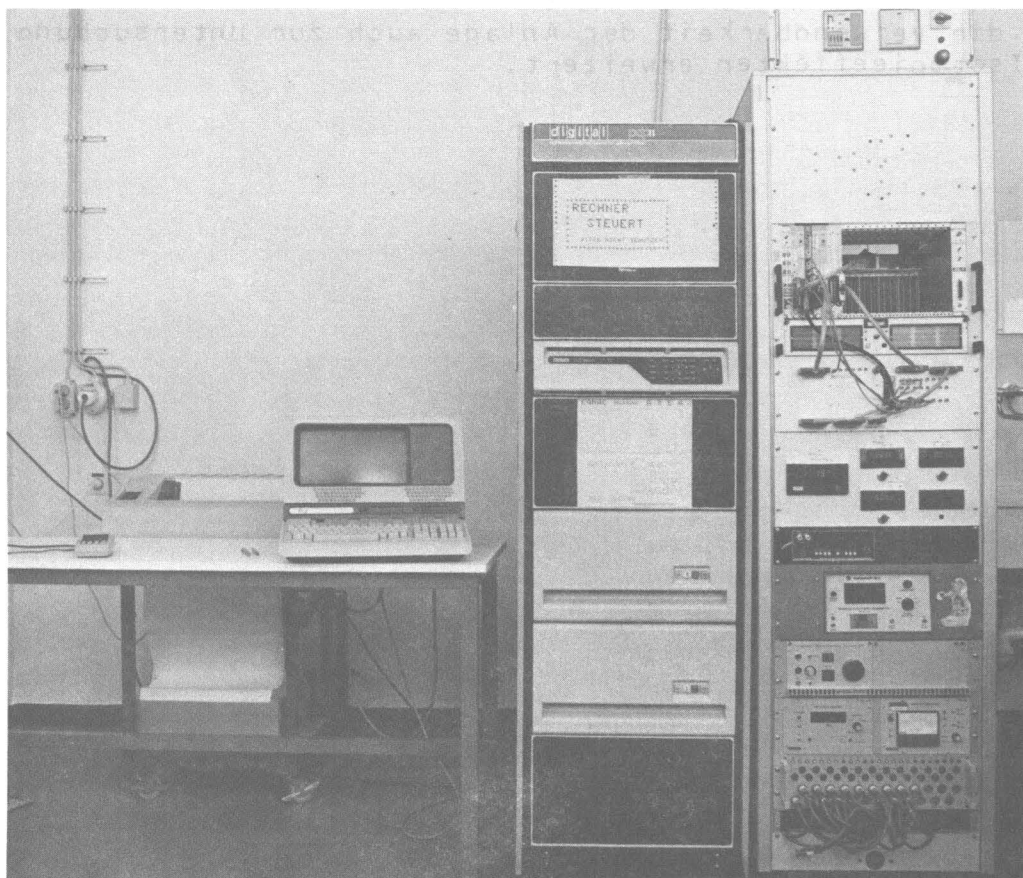
Nach Abschluss der ersten Experimente mit der aufgebauten Apparatur laesst sich bereits jetzt sagen, dass durch die Hereinnahme einer Prozessrechnersteuerung ein wesentlicher Fortschritt erzielt werden konnte. Entscheidende Vorteile einer automatisierten Apparatur ergeben sich durch folgende Faktoren:

1. Die Moeglichkeit einer Reproduzierung und Ueberpruefung von Messergebnissen ist in hohem Masse gegeben.
2. Die zeitliche Ausnutzung der Messapparatur ist in hohem Masse moeglich, sodass ansonsten sehr zeit- und Personal-intensive Messungen in vertretbarer Zeit durchfuehrbar sind. Dadurch kann die Zahl der untersuchten Proben stark erhoeht werden.
3. Durch die sofortige (teilweise graphische) Auswertung der Messergebnisse ist eine sofortige Interpretation und ein eventuell notwendiges Eingreifen des Experimentators moeglich geworden.
4. Durch die Prozessrechnersteuerung ist eine grosse Vielseitigkeit der Apparatur gegeben, wodurch dem Experimentator sowohl die Moeglichkeit der Probenpraeparation als auch die der Forschung gegeben ist.

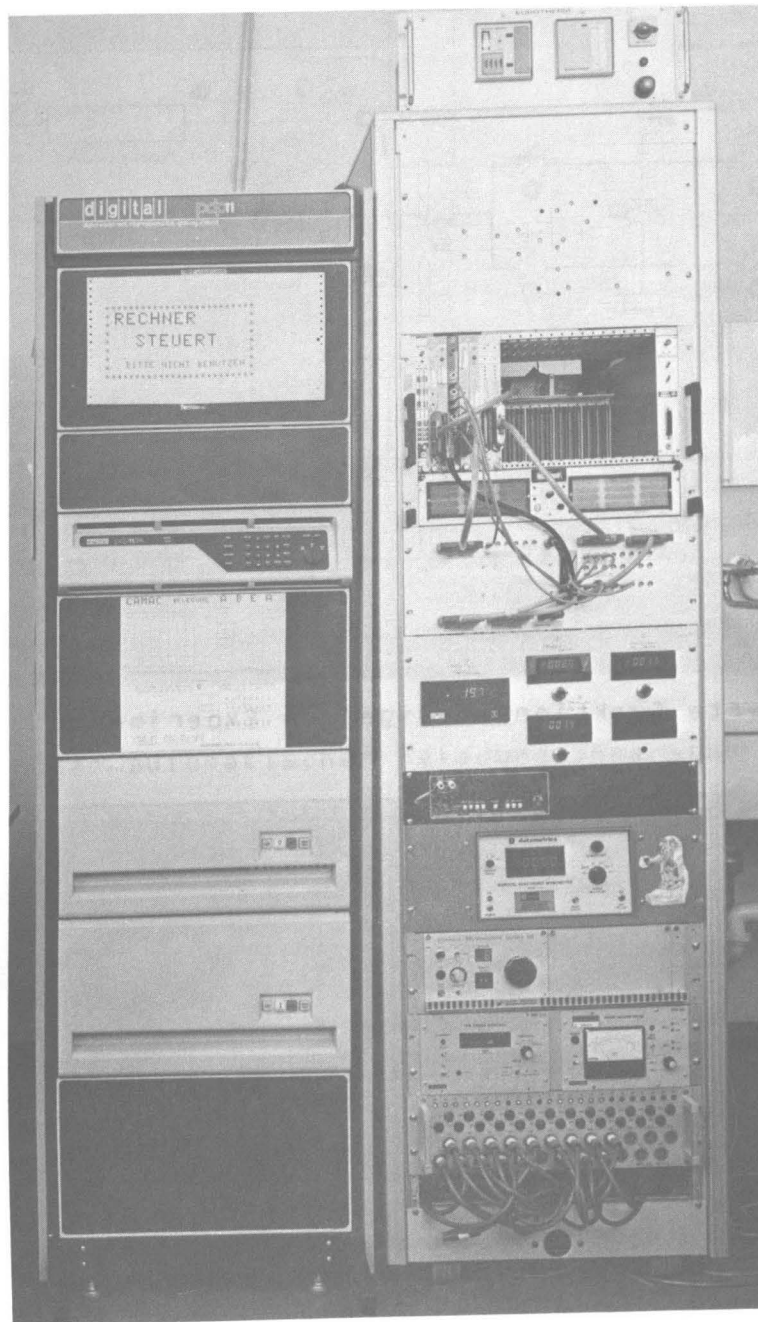
Durch den bevorstehenden Anschluss des Massenspektrometers wird die Verwendbarkeit der Anlage auch zur Untersuchung von Isotopieeffekten erweitert.



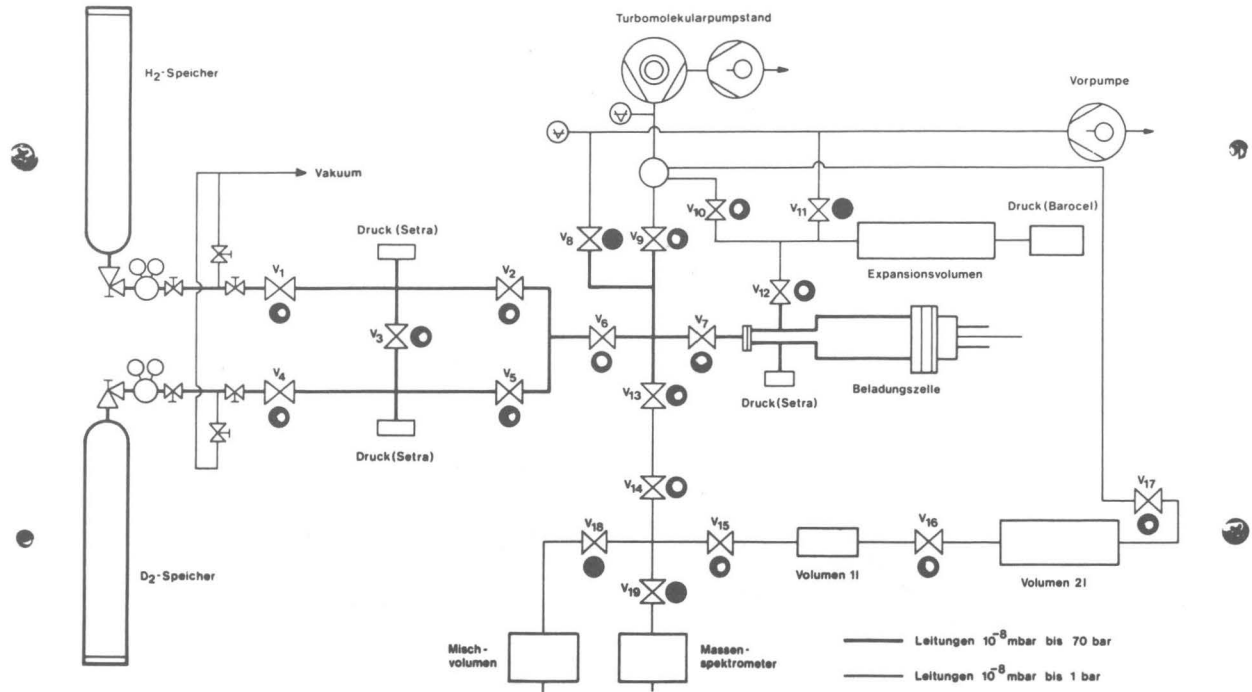
Gesamtansicht: Rechner und Experiment



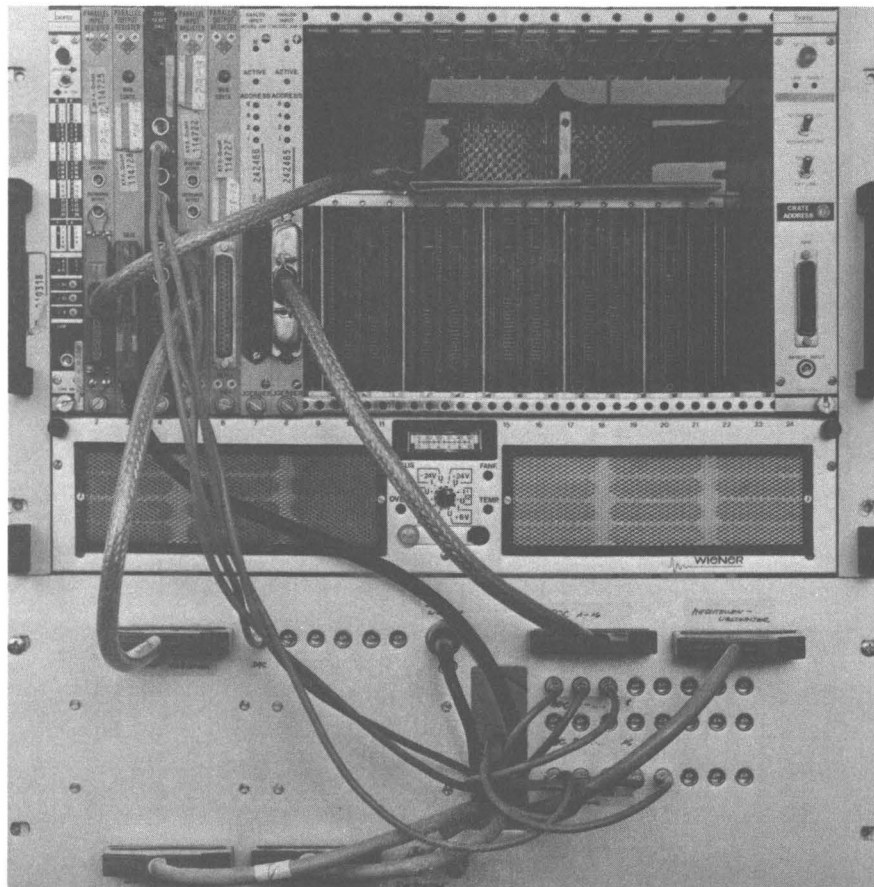
Frontansicht: Rechner und Steuerelektronik



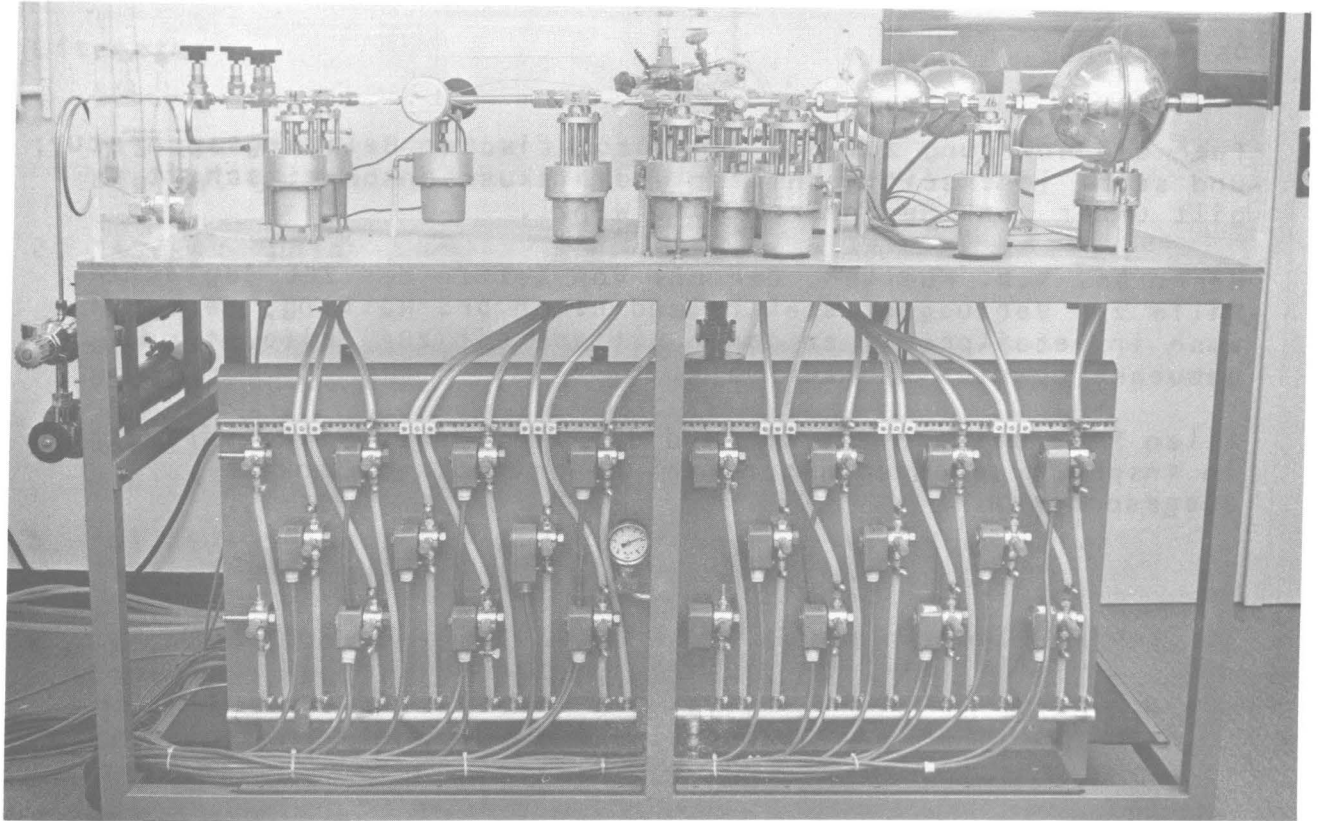
Rechner mit angeschlossener Steuerelektronik



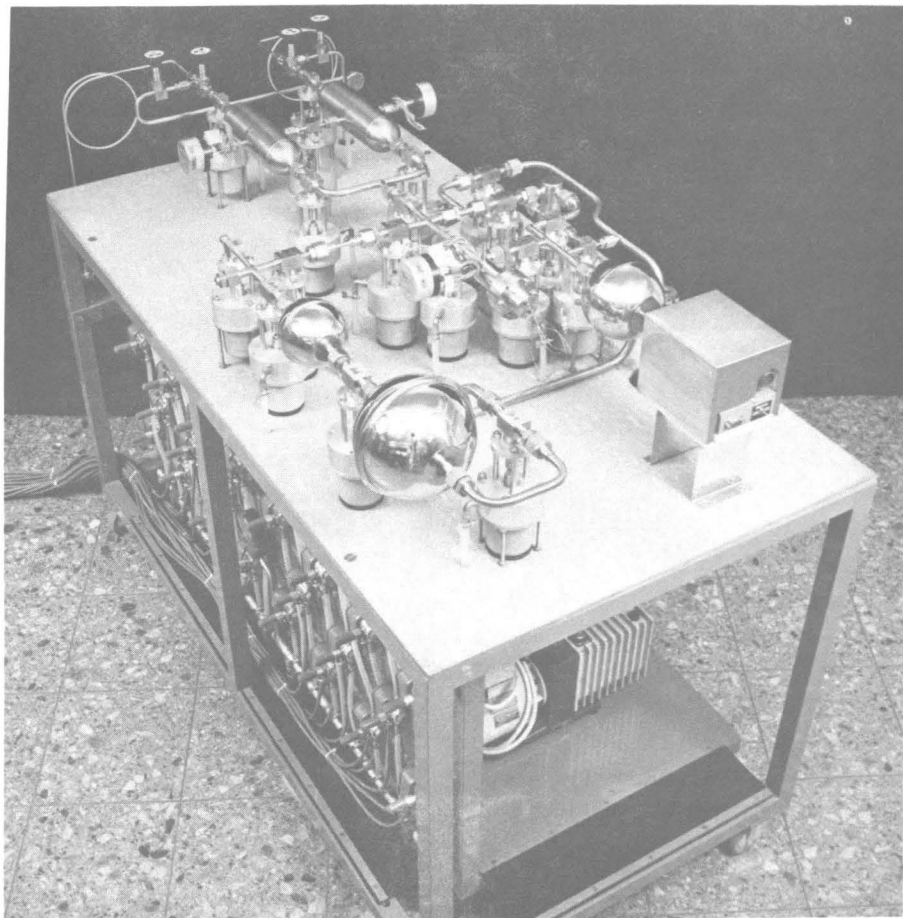
## Rechnergesteuerte Funktionsanzeige des Experiments



CAMAC - Crate mit Verkabelungsplatte



Frontansicht der automatischen Beladungsapparatur



Aufsicht auf die Apparatur

## Danksagung

Fuer die Anregung zum Bau der automatischen Beladungsapparatur und seine jederzeitige Hilfs- und Diskussionsbereitschaft gilt unser Dank Herrn Prof. Dr. Wenzl.

Herrn Dr. K.D. Mueller, der uns von seiten des ZEL jegliche Hilfe zur Verfuegung stellte und Herrn Dr. Halling, der uns auch in Detailproblemen immer hilfsbereit zur Seite stand, gebuehrt ebenfalls unser Dank.

Allen Mitarbeitern des ZEL und des IFF, deren Hilfe wir in Anspruch genommen haben, sei hiermit unser Dank ausgesprochen.



## Literatur

1. IAS/RSX FORTRAN IV Installation Guide  
Digital Equipment
2. IAS/RSX FORTRAN IV User's Guide  
Digital Equipment
3. PDP 11 FORTRAN Language Reference Manual  
Digital Equipment
4. RSX-11M Installation Guide  
Digital Equipment
5. IAS/RSX-11M RMS-11 MACRO Programmer's Reference Manual  
Digital Equipment
6. Hydrogen in Metals II, by G. Alefeld and J. Voelkl (Eds.)  
Topics in Applied Physics, Springer, Berlin (1978)
7. H. Wenzl, Persoenliche Mitteilungen